Efficient Sampling of Bernoulli-Gaussian-Mixtures for Sparse Signal Restoration

Mehdi Amrouche, Hervé Carfantan, Jérôme Idier

Efficient Sampling of Bernoulli-Gaussian-Mixtures for Sparse Signal Restoration

Mehdi Amrouche, Hervé Carfantan, Jérôme Idier, *Member, IEEE*

***Abstract*—This paper introduces a new family of prior models called *Bernoulli-Gaussian-Mixtures* (BGM), with a view to efficiently address sparse inverse problems in the Bayesian framework. The BGM family is based on continuous Location and Scale Mixtures of Gaussians (LSMG), which includes a wide range of symmetric and asymmetric heavy-tailed probability distributions. The decomposition of a distribution as a Gaussian mixture is a case of data augmentation from which we derive a Partially Collapsed Gibbs Sampler (PCGS) for the BGM, in a systematic way. the derived PCGS is shown to be more efficient than the standard Gibbs sampler, both in terms of number of iterations and CPU time. Moreover, special attention is paid to BGM involving a density defined over a real half-line. An asymptotically exact LSMG approximation is introduced, which allows us to expand the applicability of PCGS to cases such as BGM models with a non-negative support.**

***Index Terms*—Sparsity, MCMC, partially collapsed sampling, continuous Gaussian Mixtures, nonnegativity.**

I. Introduction

SPARSE signal restoration problems arise in different fields such as geophysics, astronomy and compressed sensing. The objective is to find a sparse representation $\boldsymbol{x}$ of a signal $\boldsymbol{y}$ as a linear combination of a limited number of elements (atoms) taken from a given dictionary $\mathbf{H}$. This problem is often referred to as subset selection because it consists in selecting a subset of columns of $\mathbf{H}$, so that

$$\boldsymbol{y} = \mathbf{H}\boldsymbol{x} + \boldsymbol{\epsilon}, \tag{1}$$

where $\boldsymbol{y}$ is the $N\times 1$ observed signal, $\mathbf{H}$ is a $N\times K$ matrix with $K < N$, $\boldsymbol{x}$ is a $K\times 1$ sparse signal with only $L < K$ nonzero components (corresponding to weighting coefficients), and $\boldsymbol{\epsilon}$ is a perturbation vector.

A reference method to estimate the sparse signal $\boldsymbol{x}$ consists in minimizing the squared residual error subject to a sparsity constraint:

$$\min_{\boldsymbol{x}} \|\boldsymbol{y} - \mathbf{H}\boldsymbol{x}\|^2 \ \text{ s.t. } \|\boldsymbol{x}\|_0 \leq L \tag{2}$$

where $\|\cdot\|$ and $\|\cdot\|_0$ respectively stand for the Euclidean norm and the ℓ_0 pseudo-norm. However, (2) is an NP-hard combinatorial discrete problem [1].

An alternative is the convex relaxation of problem (2) which replaces the ℓ_0 pseudo-norm by the ℓ_1 norm [2], [3], the sparsity of the solutions coming from the non-smooth character of the ℓ_1 norm at zero. Greedy algorithms, such as *Matching Pursuit* and its improved versions *Orthogonal Matching Pursuit* and *Orthogonal Least Squares* [2], [4], form another class of methods. Greedy algorithms iteratively recover the set of active atoms by incremental selections. In practice, a common difficulty shared by ℓ_0, ℓ_1 and greedy methods is that L is often an unknown quantity.

In some application fields, such as spectroscopy [5], [6] and particle image recovery [7], the signal of interest $\boldsymbol{x}$ is nonnegative, in addition to being sparse. Nonnegative adaptations have been proposed for both convex relaxation [7], [8] and greedy algorithms [9], [10].

On the other hand, we can also rely on hierarchical Bayesian models to explicitly account for sparsity. Such models incorporate an additional layer of hidden, binary components $\boldsymbol{q} = (q_k)$ to explictly encode for atom activation. Variables q_k are generally considered as independent, identically distributed (i.i.d.) according to a Bernoulli law. Amplitudes x_k are also considered as i.i.d. random variables, with a prior law defined conditionally to q_k as:

$$\begin{aligned} x_k|q_k = 1 &\sim \mathcal{D}(\boldsymbol{\theta}_x) \\ x_k|q_k = 0 &\sim \delta_0(x_k) \end{aligned} \tag{3}$$

where $\mathcal{D}(\boldsymbol{\theta}_x)$ stands for a distribution with parameters $\boldsymbol{\theta}_x$, and δ_0 is the Dirac distribution. Let us remark that the sparsity level $L = \|\boldsymbol{x}\|_0$ is governed by parameter $\xi = P(q_k = 1)$. In the sequel, priors of the form (3) are referred to as Bernoulli-$\mathcal{D}$ priors.

The posterior distribution of $(\boldsymbol{q}, \boldsymbol{x})$ can be derived thanks to the Bayes rule:

$$p(\boldsymbol{q}, \boldsymbol{x}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{q}, \boldsymbol{x}) p(\boldsymbol{x}|\boldsymbol{q}) P(\boldsymbol{q}|\xi) \tag{4}$$

Under the usual white Gaussian noise assumption $\epsilon_k \sim \mathcal{N}(0, \sigma_\epsilon^2)$, the data likelihood reads:

$$p(\boldsymbol{y}|\boldsymbol{q}, \boldsymbol{x}) = (2\pi)^{-\frac{N}{2}} \sigma_\epsilon^{-N} \exp\left(-\frac{\|\boldsymbol{y} - \mathbf{H}\boldsymbol{x}\|^2}{2\sigma_\epsilon^2}\right). \tag{5}$$

The Bernoulli-Gaussian (BG) case where the distribution $\mathcal{D}$ is Gaussian is by far the most usual choice [11]–[14]. Deterministic optimization algorithms [11], [15] and Markov chain Monte Carlo techniques (MCMC) [13], [14] have been proposed to compute the Maximum a Posteriori (MAP) and the Posterior Mean (PM) estimators, respectively.

In the MCMC framework, other priors have also been introduced:

- a Bernoulli-Laplace prior has been considered for pMRI reconstruction [16] and sparse EEG source localization [17], [18].

M. Amrouche and H. Carfantan are with the Institut de Recherche en Astrophysique et Planétologie, Université de Toulouse, CNRS/UPS/CNES, Toulouse, France. (e-mail: firstname.name@irap.omp.eu)

J. Idier is with the CNRS at the Laboratoire des Sciences du Numérique de Nantes (LS2N, CNRS UMR 6004), Nantes, France. (e-mail: jerome.idier@ls2n.fr).

- under the nonnegativity constraint, a Bernoulli-Truncated-Gaussian and a Bernoulli-Exponential have been proposed, respectively for blind spike train deconvolution [6] and sparse image reconstruction [19].

MCMC methods developed in the Bayesian framework constitute a powerful inference tool to address sparse inverse problems. For instance, in the sparse deconvolution context, where the dictionary entries are often strongly correlated, the BG prior associated to a Gibbs sampler was empirically shown to give better results than greedy algorithms and convex relaxation [20]. Moreover, in contrast with deterministic approaches, Bayesian sampling allows one to consider an unsupervised setting where the model hyper-parameters $\boldsymbol{\theta} = [\xi, \boldsymbol{\theta}_x, \sigma_\epsilon]$ are unknown and estimated jointly with the parameters of interest, or integrated out. Finally, as MCMC algorithms provide samples of the parameters distributed according to their posterior law, we can easily obtain additional statistical characteristics of interest. For instance, posterior standard deviations provide measures of confidence on the estimated quantities.

Although PM estimation of $\boldsymbol{q}$ and $\boldsymbol{x}$ obtained using MCMC methods yields satisfactory results, the computational effort may be high if basic Bayesian sampling schemes are implemented. For instance, Gibbs sampling with a site-by-site updating scheme for $(\boldsymbol{q}, \boldsymbol{x})$ (as in [6], [13], [19]) lacks efficiency, because the outcome trajectory tends to get stuck around some configurations of the Bernoulli sequence $\boldsymbol{q}$ [21], [22], leading to poor mixing properties.

In the BG prior case, improved sampling schemes have been proposed in [14]. In particular, a Partially Collapsed Gibbs Sampler (PCGS) based on [23] is proposed, which combines a step that samples $\boldsymbol{q}$ marginally with respect to $\boldsymbol{x}$ and other sampling steps involving $\boldsymbol{x}$. Compared to standard Gibbs sampling, PCGS requires far less iterations to converge. Although the computing cost of each iteration of PCGS is higher, PCGS is significantly faster than standard Gibbs to solve unsupervised sparse deconvolution problems considered in [14], [24]. Recently, Boudineau *et al.* [25] extended PCGS to problems involving dictionaries with entries that non linearly depends on additional unknown parameters. Again, the resulting extended PCGS is shown to perform better that usual Gibbs sampling.

One key condition to adopt the PCGS strategy is the possibility to marginalize the amplitudes $\boldsymbol{x}$ out of the posterior. When a BG prior is dealt with, this condition is met since the Gaussian prior $p(\boldsymbol{x}|\boldsymbol{q})$ over $\boldsymbol{x}$ is conjugate to the data likelihood (5). As a consequence, the marginal posterior distribution:

$$P(\boldsymbol{q}|\boldsymbol{y}) \propto \int p(\boldsymbol{y}|\boldsymbol{q}, \boldsymbol{x}) p(\boldsymbol{x}|\boldsymbol{q})\, d\boldsymbol{x} P(\boldsymbol{q}|\xi) \tag{6}$$

is still tractable. Unfortunately, such a useful property is not valid for any Bernoulli-$\mathcal{D}$ prior.

The aim of this paper is to reconcile PCGS sampling with several useful cases of Bernoulli-$\mathcal{D}$ priors when $\mathcal{D}$ is not Gaussian. More specifically, we will concentrate on two important cases:

(S) $\mathcal{D}$ belongs to a family of heavy-tailed, symmetric densities supported on the whole real line such as the Laplace case;

(P) $\mathcal{D}$ belongs to a family of asymmetrical densities supported on a real half-line such as the truncated Gaussian case.

A key element of our contribution is to introduce latent variables, according to a *data augmentation* principle. Akin to many previous contributions, we will rely on continuous mixtures of Gaussians (CMG) to reach our goal. Scale Mixtures of Gaussians (SMG), and Location Mixtures of Gaussians (LMG) are two well-known families of CMG [26]. In this paper, we will mainly consider Location and Scale Mixtures of Gaussians (LSMG), which is a family of CMG for which SMG are a particular case. The reason why we resort to a wider family than SMG will become clear in Sect. II-D. On the basis of LSMG decompositions, we will propose a new family of priors called *Bernoulli-Gaussian-Mixtures* (BGM).

Finally, we will devise an exact stochastic sampling scheme to deal with BGM priors, with better mixing properties than standard Gibbs sampling.

The organization of the paper is as follows. In Sect. II, we give a general overview of data augmentation based on continuous mixtures of Gaussians, with a focus on Location-Scale mixtures of Gaussians. Sect. III formally introduces the BGM prior in the Bayesian framework. The corresponding partially collapsed sampling strategy is presented in Sect. IV. In Sect. V, we empirically study the efficiency of the BGM prior and the corresponding PCGS sampler, in both (S) and (P) cases, through two sparse, unsupervised deconvolution problems. Finally, conclusions are drawn in Sect. VI.

II. Data Augmentation by Continuous Mixtures of Gaussians

A. Introduction

Inference schemes based on latent variables pertain to the *data augmentation* principle. More specifically, in the context of signal and image restoration, continuous mixtures of Gaussians (CMG) decompositions are commonly found.

Definition II.1 (CMG)**.** A random variable X is said a CMG if it can be decomposed as

$$X = M + \sqrt{W} Z, \tag{7}$$

where (M, W) is a couple of random variables and $Z \sim \mathcal{N}(0,1)$ is independent of (M, W). The joint probability measure of (M, W) is supported by $\mathbb{R} \times \mathbb{R}_+$.

The measure of (M, W) allows us to modulate the probability measure of X by tuning the mean or/and variance of the conditionally Gaussian variable $(X \mid M, W)$. Scale mixtures of Gaussians (SMG) and Location mixtures of Gaussians (LMG) correspond to cases where M or W becomes deterministic, respectively, so that only the variance or the mean of the Gaussian is modulated. Location-Scale mixtures of Gaussians (LSMG) correspond to a case where the mean and the variance are jointly modulated, as defined in Sect. II-B.

In a deterministic optimization perspective, reweighted least square algorithms [27] and half-quadratic algorithms [28], [29] can be interpreted as EM algorithms where the augmented dataset involves either an SMG or an LMG model [26].

Stochastic samplers have also been proposed based on data augmentation involving CMG models, in order to derive more efficient Gibbs samplers:

- SMG [30], [31] are more usually considered. For instance, [32] and [33] rely on an exact data augmentation scheme that corresponds to the SMG decomposition of a Laplace and a Student's t distribution, respectively.
- Some recent contributions rather consider LMG models. LMG also play a key role in the data augmentation scheme introduced in [34]. In [35], [36], more general families of location mixtures are introduced, with a special attention paid to the Gaussian case, although the connection with LMG remains implicit.
- In [37], the data augmentation scheme corresponds to the LSMG decomposition of Generalized hyperbolic (GH) variables. Recently, [38] proposed an approximate data augmentation scheme of the truncated Gaussian, also relying on the LSMG decomposition of the GH distribution.

Here, we generalize the contribution of [38] to the entire LSMG family, allowing one to consider heavy-tailed, possibly asymmetric, distributions.

B. Location-Scale Mixtures of Gaussians (LSMG)

Definition II.2 (LSMG)**.** A random variable X is a location-scale mixture of Gaussians if

$$X = \mu + \beta W + \sqrt{W} Z \tag{8}$$

where $\beta,\ \mu \in \mathbb{R}$, $Z \sim \mathcal{N}(0, \sigma_z^2)$ and $W > 0$ being independent random variables, with $\sigma_z > 0$. W will be called the mixing variable of X.

LSMG have been introduced in [39] under the name of *normal variance-mean mixtures* (see also [40]). LSMG are CMG since the normalized version of (8) identifies with (7) when $M = \mu + \beta W$. LSMG have been used in financial [41], [42], and statistical data analysis [43], [44] applications, for their ability to model asymmetric and/or heavy-tailed distributions. In particular, the Generalized-Hyperbolic family introduced in [45] is a subfamily of LSMG distributions that encompasses several cases of interest [46].

Table I gives some typical examples of LSMG distributions. Note that both AL and VG cases are limit cases of the GH family corresponding to $GH(1, \alpha, \beta, 0, \mu)$ and $GH(\lambda, \alpha, \beta, 0, \mu)$, respectively. In all cases, μ is a location parameter, while β tunes the skewness of the density.

In the next two subsections, we examine the (S) and (P) cases more carefully, the latter being substantially more complex.

C. S Case: Heavy-Tailed, Symmetric LSMG

For $\beta = 0$, (8) defines a symmetric random variable about μ. Indeed, symmetric LSMG boil down to the family of shifted SMG. In Sect. V-B, we consider the case of Laplace distribution in the framework of the BGM model proposed in Section III, to illustrate that PCGS provides an improved sampler in such a situation.

TABLE I
EXAMPLES OF LSMG DISTRIBUTIONS ($\gamma^2 = \alpha^2 - \beta^2$ AND $\sigma_z = 1$)

Distribution of X	Distribution of W
Asymmetric Laplace (AL) [47, §3] $AL(\alpha, \beta, \mu)$	Exponential $E\left(\frac{\gamma^2}{2}\right)$
Variance-Gamma (VG) [47, §4.1] $VG(\lambda, \alpha, \beta, \mu)$	Gamma $\Gamma\left(\lambda, \frac{\gamma^2}{2}\right)$
Generalized-Hyperbolic (GH) [45] $GH(\lambda, \alpha, \beta, \delta, \mu)$	Generalized-Inverse-Gaussian [48] $GIG(\lambda, \gamma, \delta)$

D. P Case: LSMG Approximations of Densities on $\mathbb{R}_+$

As stated above, CMG models cover a wide range of probability distributions. However, an obvious restriction is that the density of any CMG is supported on the whole real line. In the case of densities only defined over real half-lines, our key idea is to rely on LSMG *approximations* instead of LSMG *decompositions*. For the sake of simplicity, we will only deal with the case of $\mathbb{R}_+$, which is obtained by considering (8) with $\mu = 0$ and $\beta > 0$. Symmetrically, approximations of densities on $\mathbb{R}_-$ will be obtained with $\mu = 0$ and $\beta < 0$, and in both cases, shifted versions can be built by considering non-zero values of μ.

Let us consider a target density p^* defined on $\mathbb{R}_+$, that we would like to approximately decompose as an LSMG. We have the following proposition.

Proposition II.1. *Let X be an LSMG defined according to* (8) *with $\mu = 0$, $\beta > 0$, using a mixing variable W with a pdf*

$$q_\beta(w) = \beta\, p^*(\beta w). \tag{9}$$

We have the following properties:

(i) When $\beta \to \infty$, the LSMG X converges in probability towards $X^* = \beta W$, whose pdf is the target p^*.

(ii) If the first two moments of X^* exist, then

$$\mathrm{E}[X] = \mathrm{E}[X^*] > 0, \tag{10}$$

$$\mathrm{var}(X) = \mathrm{var}(X^*) + \frac{\sigma_z^2}{\beta}\mathrm{E}[X^*]\,, \tag{11}$$

$$\mathrm{Corr}(X, W) = \mathrm{Corr}(X, X^*) = \left(1 + \frac{\sigma_z^2 \mathrm{E}[X^*]}{\beta\, \mathrm{var}(X^*)}\right)^{-\frac{1}{2}}, \tag{12}$$

where Corr *denotes the Pearson correlation coefficient.*

Proof: See Appendix A. ∎

Let us denote p_β the pdf of the LSMG X obtained along the lines of Proposition II.1. In view of (i), we will call p_β an *asymptotically Exact Location-Scale Approximation* (ELSA) of the density p^*. Parameter β will allow us to tune the quality of approximation, and (10)-(11) indicate that for any fixed β, p_β can be viewed as a diffuse approximation of p^*. In terms of approximation quality, large values of β are clearly optimal. However, in terms of sampling efficiency, the interest of such a data augmentation scheme is annihilated for large

values of β since (12) implies that the larger β is, the more X and W are correlated. In practice, we must avoid too large values of β, and control the trade-off between the quality of the approximation and the efficiency of data augmentation.

Fig. 1 gives examples of the ELSA density p_β when the target pdf p^* is a truncated Gaussian density on $\mathbb{R}_+$ [49]. In this case, we will denote the ELSA density $p_\beta \sim NTG(0, \sigma_z, \beta)$ for *Normal-Trucated-Gaussian*, with parameters β and σ_z.

In comparison, we have also considered an LMG approximation $X' = X^* + \rho Z$, whose density p'_ρ is simply a smoothed version of p^*. Note that p'_ρ is an *Asymptotically exact data augmentation* (AXDA) model, in the terminology of [36]. According to Proposition II.1(i), the ELSA model is also an AXDA model, although not considered in [36]. AXDA approximations of the LMG type share similar properties with Proposition II.1:

- p'_ρ converges in probability towards p^* when $\rho \to 0$;
- $\mathrm{E}[X'] = \mathrm{E}[X^*]$, $\mathrm{var}(X') = \mathrm{var}(X^*) + \rho^2\sigma_z^2$;
- $\mathrm{Corr}(X', X^*) = \left(1 + \frac{\rho^2\sigma_z^2}{\mathrm{var}(X^*)}\right)^{-\frac{1}{2}}$.

Nonetheless, some features of the truncated Gaussian target are better preserved with ELSA: $\forall \rho > 0$, p'_ρ is smooth at zero, while $\forall \beta > 0$, p_β preserved the nonsmoothness of p^* at zero. Moreover, p_β reaches its maximum at zero and is monotonically decreasing on $\mathbb{R}_+$. For these reasons, we made the choice to build our PCGS strategy upon LSMG rather than LMG models. However, it would remain possible to develop a similar PCGS strategy on the basis of LMG models. The latter could be of interest when it happens difficult to define and manipulate LSMG approximations, such as the case of densities restricted to a finite interval.

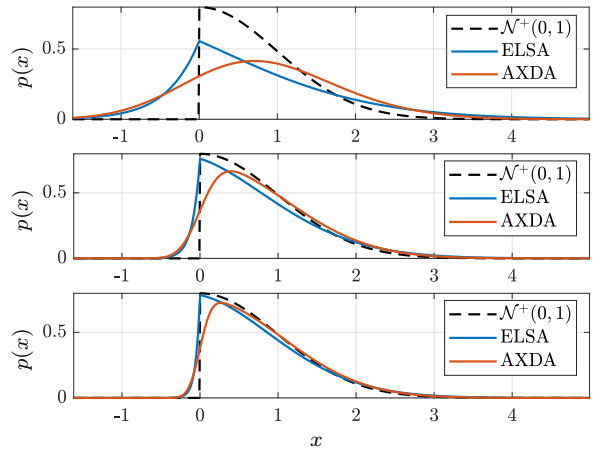

Fig. 1. Examples of the ELSA pdf $p_\beta \sim NTG(0, 1, \beta)$ (blue line) when $\sigma_z^2 = 1$ and p^* is the standard truncated Gaussian pdf $\mathcal{N}^+(0, 1)$ (black dashed line). From top to bottom, β is set to 1, 3 and 9, respectively. For the sake of comparison, the red curve corresponds to the pdf of an LMG (coined AXDA in [36]) of the same approximation quality as p_β in the sense of the total variation.

Finally, let us examine the scale invariance property of ELSA approximations. This is an important matter in practice since in many realistic applications to signal restoration, the scale of the unknown signals is also unknown and must be sampled within the MCMC framework. In such a situation, we need an ELSA model with a quality of approximation that does not fluctuate with the value of the scale parameter.

Let $X_s^* = sX^*$ and $W_s = sW$ be scaled versions of X^* and W, for a given $s > 0$. Let us define the LSMG

$$X_s = \beta W_s + \sqrt{W_s} Z_s \tag{13}$$

with

$$Z_s = \sqrt{s} Z \sim \mathcal{N}(0, s\sigma_z^2). \tag{14}$$

Then, $X_s = s(\beta W + \sqrt{W} Z) = sX$, and we have the following property.

Proposition II.2 (Scale invariance)**.** *Let p_s^* and $p_{\beta,s}$ denote the pdf of X_s^* and X_s, respectively. The total variation of the approximation error $p_{\beta,s} - p_s^*$ does not depend on the scale s:*

$$\int_{\mathbb{R}} |p_{\beta,s}(x) - p_s^*(x)|\, dx = \int_{\mathbb{R}} |p_\beta(x) - p^*(x)|\, dx. \tag{15}$$

Moreover:

$$\int_{-\infty}^{0} p_{\beta,s}(x)\, dx = \int_{-\infty}^{0} p_\beta(x)\, dx. \tag{16}$$

Proof: Identities (15)-(16) straightforwardly derive from the fact that both $X_s = sX$ and $X_s^* = sX^*$ hold. ∎

Let us remark that we need to modulate the variance of the generating Gaussian in an appropriate way to get a scale invariance property on the approximation error. We could not get a similar result using a Gaussian variable with a normalized variance.

III. Bernoulli-Gaussian-Mixture model

A. Prior Distribution

Definition III.1. A Bernoulli-Gaussian-Mixture (BGM) is a Bernoulli-$\mathcal{D}$ model when $\mathcal{D}$ is an LSMG:

- $\boldsymbol{q}$ is distributed according to an independent Bernoulli law $\mathcal{B}(\xi)$: for any vector $\boldsymbol{q} \in \{0,1\}^K$,
$$P(\boldsymbol{q}; \xi) = \prod_{k=1}^{K} \xi^k (1-\xi)^{1-k};$$
- $\boldsymbol{w}$ is an independent random vector of variable length, each variable w_k being defined when $q_k = 1$ only, according to a pdf $p_W(\cdot; \boldsymbol{\theta}_w)$ on $\mathbb{R}_+$;
- $\boldsymbol{x}$ is an independent random vector of length K defined conditionally to $\boldsymbol{q}$ and $\boldsymbol{w}$ as:
$$\begin{aligned} &\text{if } q_k = 0, \quad x_k = 0, \\ &\text{if } q_k = 1, \quad x_k | w_k, s \sim \mathcal{N}(s\beta w_k, s^2 w_k); \end{aligned}$$

so that $x_k | (q_k = 1)$ is distributed according to an LSMG distribution with $\mu = 0$. The latter is driven by the skewness parameter β, the scale $s > 0$ and the shape parameters of $p_W(\cdot; \boldsymbol{\theta}_w)$.

Distribution $\mathcal{D}$ is an LSMG model that corresponds either to an (S) (for $\beta = 0$) or a (P) (for $\beta > 0$) case according to Sect. II-C and II-D, respectively. Note that we introduced a scale parameter s that will be considered unknown, while parameter β (which controls the degree of approximation

between $\mathcal{D}$ and a target distribution $\mathcal{D}^*$ with ELSA) will be considered fixed. On the other hand, we will also consider $\boldsymbol{\theta}_w$ to be known, for the sake of simplicity. Assuming that $\boldsymbol{\theta}_w$ is unknown would naturally add a sampling step in the considered MCMC algorithm. In [37], the design of such a step when W is a GIG is examined in detail.

Finally, note that we have fixed $\sigma_z = 1$ in (8), so that $x_k|w_k \sim \mathcal{N}(\beta w_k, w_k)$ when $s = 1$. In other words, the adopted LSMG model matches the usual definition (involving a normalized Gaussian) when (*and only when*) the scale is unity, which is a natural choice.

B. Posterior Distribution

The additive noise $\boldsymbol{\epsilon}$ is considered Gaussian and zero-mean, with covariance $\boldsymbol{\Gamma}_\epsilon$. The posterior distribution reads:

$$p(\boldsymbol{q},\boldsymbol{x},\boldsymbol{w},\boldsymbol{\theta}|\boldsymbol{y}) \propto |\boldsymbol{\Gamma}_\epsilon|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\left\|\boldsymbol{y}-\overline{\overline{\mathbf{H}}}\overline{\boldsymbol{x}}\right\|^2_{\boldsymbol{\Gamma}_\epsilon^{-1}}\right) \times p(\boldsymbol{x},\boldsymbol{w}|\boldsymbol{q},\boldsymbol{\theta})P(\boldsymbol{q}|\xi)p(\boldsymbol{\theta}) \quad (17)$$

where $\propto$ denotes the proportionality sign, $\overline{\boldsymbol{x}}$ and $\overline{\overline{\mathbf{H}}}$ respectively gather the entries x_k and the columns $\boldsymbol{h}_k$ for which $q_k = 1$, $\boldsymbol{\theta} = \{\xi, \boldsymbol{\Gamma}_\epsilon, s\}$ are the hyper-parameters to be sampled, and $\|\boldsymbol{v}\|^2_{\mathbf{A}} = \boldsymbol{v}^{\mathrm{t}}\mathbf{A}\boldsymbol{v}$.

C. Partially Marginalized Posterior Distribution

From (17), one can deduce that $\overline{\boldsymbol{x}}|\boldsymbol{q},\boldsymbol{w},\boldsymbol{y},\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Gamma}_1)$, with

$$\boldsymbol{\mu}_1 = \mathrm{E}\big[\overline{\boldsymbol{X}} \,|\, \boldsymbol{y},\boldsymbol{q},\boldsymbol{w},\boldsymbol{\theta}\big] = \boldsymbol{\Gamma}_1\left(\overline{\overline{\mathbf{H}}}^{\mathrm{t}}\boldsymbol{\Gamma}_\epsilon^{-1}\boldsymbol{y} + \boldsymbol{\Gamma}_0^{-1}\boldsymbol{\mu}_0\right), \quad (18)$$

$$\boldsymbol{\Gamma}_1 = \mathrm{cov}\big(\overline{\boldsymbol{X}} \,|\, \boldsymbol{y},\boldsymbol{q},\boldsymbol{w},\boldsymbol{\theta}\big) = \left(\overline{\overline{\mathbf{H}}}^{\mathrm{t}}\boldsymbol{\Gamma}_\epsilon^{-1}\overline{\overline{\mathbf{H}}} + \boldsymbol{\Gamma}_0^{-1}\right)^{-1}, \quad (19)$$

where

$$\boldsymbol{\mu}_0 = \mathrm{E}\big[\overline{\boldsymbol{X}} \,|\, \boldsymbol{q},\boldsymbol{w},\boldsymbol{\theta}\big] = s\beta\boldsymbol{w},$$
$$\boldsymbol{\Gamma}_0 = \mathrm{cov}\big(\overline{\boldsymbol{X}} \,|\, \boldsymbol{q},\boldsymbol{w},\boldsymbol{\theta}\big) = s^2\mathbf{W},$$

with $\mathbf{W} = \mathrm{diag}\{\boldsymbol{w}\}$. As a consequence, the marginalized posterior of ($\boldsymbol{q}$,$\boldsymbol{w}$,$\boldsymbol{\theta}$) with respect to $\boldsymbol{x}$ reads

$$p(\boldsymbol{w},\boldsymbol{\theta}|\boldsymbol{q},\boldsymbol{y})P(\boldsymbol{q}|\boldsymbol{\theta},\boldsymbol{y}) \propto |\mathbf{B}|^{-\frac{1}{2}}\ P(\boldsymbol{q}|\xi)p(\boldsymbol{\theta})\prod_k p_W(w_k;\boldsymbol{\theta}_w) \times \exp\left(-\frac{1}{2}\big(\boldsymbol{y}^{\mathrm{t}}\mathbf{B}^{-1}\boldsymbol{y} + \boldsymbol{\mu}_0^{\mathrm{t}}\mathbf{C}^{-1}\boldsymbol{\mu}_0 - 2\boldsymbol{y}^{\mathrm{t}}\mathbf{D}\boldsymbol{\mu}_0\big)\right)$$

where

$$\mathbf{B}^{-1} = \boldsymbol{\Gamma}_\epsilon^{-1} - \boldsymbol{\Gamma}_\epsilon^{-1}\overline{\overline{\mathbf{H}}}\boldsymbol{\Gamma}_1\overline{\overline{\mathbf{H}}}^{\mathrm{t}}\boldsymbol{\Gamma}_\epsilon^{-1}, \quad (20)$$

$$\mathbf{C}^{-1} = \boldsymbol{\Gamma}_0^{-1} - \boldsymbol{\Gamma}_0^{-1}\boldsymbol{\Gamma}_1\boldsymbol{\Gamma}_0^{-1}, \quad (21)$$

$$\mathbf{D} = \boldsymbol{\Gamma}_\epsilon^{-1}\overline{\overline{\mathbf{H}}}\boldsymbol{\Gamma}_1\boldsymbol{\Gamma}_0^{-1}, \quad (22)$$

are $N \times N$, $L \times L$, and $N \times L$, respectively, with $L = \sum_k q_k$.

D. Hyper-Parameter Sampling

In order to generate samples of the hyper-parameter vector $\boldsymbol{\theta} = (\xi, \boldsymbol{\Gamma}_\epsilon, s)$, a prior distribution $p(\boldsymbol{\theta})$ needs to be defined. Firstly, we naturally suppose that ξ, $\boldsymbol{\Gamma}_\epsilon$, s are independent.

- We choose an uninformative conjugate prior $\mathcal{B}e(1,1)$ (*i.e.*, a Beta distribution) for ξ, such that its posterior distribution is given by $\xi \sim \mathcal{B}e(L+1, K-L+1)$.
- For the sake of simplicity, we only deal with the white noise case, *i.e.*, $\boldsymbol{\Gamma}_\epsilon = \sigma_\epsilon^2\mathbf{I}_N$, where $\mathbf{I}_N$ is the identity matrix of size N, so that we only need to sample the scalar parameter σ_ϵ^2. We then consider an uninformative conjugate prior $IG(1,1)$ (*i.e.*, Inverse-Gamma distribution) such that its posterior is $IG\left(\frac{N}{2}+1, \frac{1}{2}\|\boldsymbol{y}-\mathbf{H}\boldsymbol{x}\|^2 + 1\right)$. Note that handling more complex noise covariance structures would have no impact on the other sampling steps.
- We consider an Inverse-Gamma uninformative prior $s^2 \sim IG(1,1)$ for the scale parameter. In the (S) case, the chosen prior is conjugate and the posterior is $s^2 \sim IG\left(L+1, \frac{1}{2}\|\overline{\boldsymbol{x}}\|^2_{\mathbf{W}^{-1}} + 1\right)$. In the (P) case, the posterior is not a standard law. A random walk Metropolis-Hastings step with a truncated Gaussian proposal will then be used within the PCGS.

Finally, let us mention that hyper-parameters ξ and σ_ϵ^2 are independent of $\boldsymbol{w}$ and s given $\boldsymbol{y},\boldsymbol{q},\boldsymbol{x}$, so their sampling does not depend on s, β, and p_W.

IV. PCGS for the BGM Model

The proposed PCGS sampling strategy is summarized in Algorithm 1. At each iteration, first sample q_k and w_k for each k marginally with respect to $\boldsymbol{x}$, then sample $\boldsymbol{x}$ and finally the hyper-parameters $\boldsymbol{\theta}$ from their posterior.

Algorithm 1: Proposed PCGS algorithm

At each iteration t:
1) for all k in $1,\ldots,K$, draw $q_k, w_k|\boldsymbol{q}_{-k},\boldsymbol{w}_{-k},\boldsymbol{\theta},\boldsymbol{y}$ according to Algorithm 2
2) draw $\boldsymbol{x}|\boldsymbol{q},\boldsymbol{w},\boldsymbol{\theta},\boldsymbol{y}$ (Gaussian distribution)
3) draw $\boldsymbol{\theta}|\boldsymbol{q},\boldsymbol{w},\boldsymbol{x},\boldsymbol{y}$

While Step 2 is a fairly easy task since $\overline{\boldsymbol{x}}$ is a Gaussian vector of size L, Step 1 is more complex. To begin with, each w_k is defined only if $q_k = 1$, so that the posterior distribution is defined in a space with a varying dimension. Reversible-Jump (RJ) MCMC methods [50] are suited to manage jumps between subspaces of different dimensions. For the problem considered here, two states can be distinguished; whether $q_k = 1$ and $w_k \in \mathbb{R}^+$, or $q_k = 0$ and w_k is not defined. The RJ-MCMC framework allows to jump between these two states using the following moves:

- Birth: from state $q_k = 0$, propose $(q'_k = 1, w'_k)$,
- Death: from state $(q_k = 1, w_k)$, propose $q'_k = 0$,
- Update: from state $(q_k = 1, w_k)$, propose $(q'_k = 1, w'_k)$.

Let us denote $p_{uu'}$ the probability of proposing a move from the state u to u'. Since we systematically propose a *birth* move when $q_k = 0$ then $p_{01} = 1$. Otherwise, when $q_k = 1$, it is reasonable to randomly propose either a *death* or an *update* move with equal probabilities $p_{10} = p_{11} = \frac{1}{2}$.

The w'_k candidates are chosen according to the following proposal distributions.

- Birth: w'_k is sampled according to its prior density, so $q_{01}(w'_k) = p_W(w'_k;\boldsymbol{\theta}_w)$.
- Death: the proposal is deterministic and w_k is no longer defined.

Algorithm 2: Sampling q_k and w_k using the RJ framework

if $q_k = 0$ **then**
- **birth:**
  - propose $q'_k = 1$ and $w'_k \sim q_{01}(w'_k)$
  - accept with probability α_{01}.

else
- **death** (with probability p_{10}):
  - propose $q'_k = 0$
  - accept with probability α_{10}.
- **update** (with probability $p_{11} = 1 - p_{10}$):
  - propose $q'_k = 1$ and $w'_k \sim q_{11}^{(i)}(w'_k)$ where $i \in \{1, 2\}$ with equiprobability
  - accept with probability $\alpha_{11}^{(i)}$.

end

- Update: a mix between two proposals is considered. The first one is made according to the prior density $q_{11}^{(1)}(w'_k) = p_W(w'_k; \boldsymbol{\theta}_w)$, allowing a better exploration of $\mathbb{R}_+$. The second one performs a local exploration of the posterior according to a random-walk Metropolis-Hastings scheme [51], [52], with a proposal density $q_{11}^{(2)}$ corresponding to a truncated Gaussian $\mathcal{N}^+(w_k, \sigma_w^2)$. Parameter σ_w is tuned to ensure an optimal acceptance rate, a common choice being 30%. The reader may refer to [53] for a review on optimal scaling of Metropolis-Hastings algorithms.

The resulting RJ-MCMC step is summarized in Algorithm 2. To ensure the reversibility property, and thus the invariance of the Markov chain with respect to the posterior distribution [50], [54], the candidates are accepted according to $\alpha_{uu'} = \min\{1, r_{uu'}\}$, where

$$
\begin{aligned}
r_{01} &= \frac{p(\boldsymbol{w}'|\boldsymbol{q}', \boldsymbol{\theta}, \boldsymbol{y}) P(\boldsymbol{q}'|\boldsymbol{\theta}, \boldsymbol{y})\, p_{10}}{p(\boldsymbol{w}|\boldsymbol{q}, \boldsymbol{\theta}, \boldsymbol{y}) P(\boldsymbol{q}|\boldsymbol{\theta}, \boldsymbol{y})\, p_{01}\, q_{01}(w'_k)}, \\
r_{10} &= \frac{p(\boldsymbol{w}'|\boldsymbol{q}', \boldsymbol{\theta}, \boldsymbol{y}) P(\boldsymbol{q}|\boldsymbol{\theta}, \boldsymbol{y})\, p_{01}\, q_{01}(w'_k)}{p(\boldsymbol{w}|\boldsymbol{q}, \boldsymbol{\theta}, \boldsymbol{y}) P(\boldsymbol{q}|\boldsymbol{\theta}, \boldsymbol{y})\, p_{10}}, \\
r_{11}^{(1)} &= \frac{p(\boldsymbol{w}'|\boldsymbol{q}', \boldsymbol{\theta}, \boldsymbol{y}) P(\boldsymbol{q}'|\boldsymbol{\theta}, \boldsymbol{y}) q_{11}^{(1)}(w_k)}{p(\boldsymbol{w}|\boldsymbol{q}, \boldsymbol{\theta}, \boldsymbol{y}) P(\boldsymbol{q}|\boldsymbol{\theta}, \boldsymbol{y}) q_{11}^{(1)}(w'_k)}, \\
r_{11}^{(2)} &= \frac{p(\boldsymbol{w}'|\boldsymbol{q}', \boldsymbol{\theta}, \boldsymbol{y}) P(\boldsymbol{q}'|\boldsymbol{\theta}, \boldsymbol{y})}{p(\boldsymbol{w}|\boldsymbol{q}, \boldsymbol{\theta}, \boldsymbol{y}) P(\boldsymbol{q}|\boldsymbol{\theta}, \boldsymbol{y})} \frac{\eta(w_k)}{\eta(w'_k)}
\end{aligned}
$$

with $\eta(w) = 1 + \operatorname{erf}\left(\frac{w}{\sqrt{2}\sigma_w}\right)$.

The latter expressions can be made more explicit according to:

$$
r_{01} = \exp\left(\chi_{01} - \frac{1}{2}\left(\psi(\mathbf{B}_{(1)}, \mathbf{B}) + \phi_1 - \phi\right)\right) \tag{23}
$$

$$
r_{10} = \exp\left(\chi_{10} - \frac{1}{2}\left(\psi\left(\mathbf{B}_{(0)}, \mathbf{B}\right) + \phi_0 - \phi\right)\right) \tag{24}
$$

$$
r_{11}^{(1)} = \exp\left(-\frac{1}{2}\left(\psi\left(\mathbf{B}_{(1)}, \mathbf{B}\right) + \phi_1 - \phi\right)\right) \tag{25}
$$

$$
r_{11}^{(2)} = r_{11}^{(1)} \frac{p_W(w'_k; \boldsymbol{\theta}_w)}{p_W(w_k; \boldsymbol{\theta}_w)} \frac{\eta(w_k)}{\eta(w'_k)} \tag{26}
$$

with:

$$
\chi_{uu'} = \ln\left(\frac{P(q_k = u'|\xi)}{P(q_k = u|\xi)} \frac{p_{u'u}}{p_{uu'}}\right) \tag{27}
$$

$$
\psi(\mathbf{B}, \mathbf{B}') = \ln|\mathbf{B}| - \ln|\mathbf{B}'| + \boldsymbol{y}^{\mathrm{t}}\left((\mathbf{B}')^{-1} - \mathbf{B}^{-1}\right)\boldsymbol{y} \tag{28}
$$

$$
\phi = \boldsymbol{\mu}_0^{\mathrm{t}} \mathbf{C}^{-1} \boldsymbol{\mu}_0 - 2\boldsymbol{y}^{\mathrm{t}} \mathbf{D} \boldsymbol{\mu}_0 \tag{29}
$$

$$
\phi_u = \boldsymbol{\mu}_{0(u)}^{\mathrm{t}} \mathbf{C}_{(u)}^{-1} \boldsymbol{\mu}_{0(u)} - 2\boldsymbol{y}^{\mathrm{t}} \mathbf{D}_{(u)} \boldsymbol{\mu}_{0(u)} \tag{30}
$$

where for a given matrix (or vector) $\mathbf{A}$, $\mathbf{A}_{(0)}$ and $\mathbf{A}_{(1)}$ respectively correspond to $\mathbf{A}$ with a modification implied by

- $q'_k = 0$ in position k,
- $q'_k = 1$ and w'_k in position k.

Note that function ψ implicitly depends on $\boldsymbol{y}$, while matrices $\mathbf{B}$ and $\mathbf{C}$ depend on $\boldsymbol{q}$ and $\boldsymbol{w}$ through matrix $\boldsymbol{\Gamma}_1$ according to (20), (21) and (19).

A direct implementation of Algorithm 2 using (23)-(26) would require to compute many matrix inverses and determinants, which would be inefficient. We rather propose to extend the recursive implementation technique introduced in [14]. Under the usual Bernoulli-Gaussian prior considered therein, $\boldsymbol{\mu}_0 = \mathbf{0}$ and the covariance matrix $\boldsymbol{\Gamma}_0$ is proportional to identity (which means that matrices $\mathbf{C}$ and $\mathbf{D}$ are irrelevant). We propose an efficient implementation that extends [14] to any diagonal covariance matrix $\boldsymbol{\Gamma}_0$ and expectation vectors $\boldsymbol{\mu}_0$. A key point is to recursively handle Cholesky factors instead of matrices $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ (more details are given in Appendix B).

Let us emphasize that the cost per iteration of the PCGS is highly dependent on the efficient computation of the ratios r_{uu}, which crucially depend on that of function ψ. Clearly, the direct computation of ψ according to (28) is inefficient, as the latter requires to invert matrix $\mathbf{B}^{-1}$ and to evaluate $|\mathbf{B}|$, which scales as $\mathcal{O}(N^3)$. Thanks to the mathematical simplifications given in Appendix B-A, the computation of ψ boils down to the evaluation of scalars ρ'_{q_k} and γ'_{q_k}, given by (41) and (42), which scales in the most general case (*i.e.*, no favorable structure for the noise covariance $\boldsymbol{\Gamma}_\epsilon$ and for the dictionary $\mathbf{H}$) as $\mathcal{O}(N^2)$. Moreover, the computational complexity can be further reduced in some favorable situations:

- when the noise is assumed independent (*i.e.*, $\boldsymbol{\Gamma}_\epsilon$ is diagonal), (41) and (42) scale as $\mathcal{O}(NL)$;
- when the noise is assumed i.i.d. and $\mathbf{H}^{\mathrm{t}}\mathbf{H}$ and $\mathbf{H}\boldsymbol{y}$ can be computed and stored once for all (as is the deconvolution case of Sect. V-B), the computational complexity of (41) and (42) can be brought down to $\mathcal{O}(L^2)$ as described in Appendix B-F;
- when the noise is assumed i.i.d. and $\mathbf{H}$ is a sparse matrix (as is the semi-blind deconvolution case of Sect. V-C), (41) and (42) scale as $\mathcal{O}(L^2)$ using the strategy of Appendix B-F, considering that the on-the-fly computation of $\mathbf{H}^{\mathrm{t}}\mathbf{H}$ and $\mathbf{H}\boldsymbol{y}$ is inexpensive.

Furthermore, thanks to the simplifications of Appendix B-D, computation of function ϕ does not require that of matrices $\mathbf{C}^{-1}$ and $\mathbf{D}$, and its cost is lower than that of ψ. Finally, in the (S) case, function ϕ is irrelevant as $\boldsymbol{\mu}_0 = \mathbf{0}$.

V. Numerical Validations

A. Simulation framework

In order to compare the practical efficiency of the proposed PCGS sampler to that of a standard Gibbs strategy, we propose a couple of experiments. The first is a sparse deconvolution problem under a Bernoulli-Laplace prior [16], [18], which is an example of a heavy-tailed, symmetric density as dealt in Sect. II-C. In the second one, we consider in addition a nonnegativty constraint that is handled thanks to a Bernoulli-Truncated-Gaussian prior [6], so that we are now in the situation dealt in Sect. II-D.

To empirically compare the computing time of the different samplers, we make use of Brooks and Gelman's graphical method to assess convergence [55]. It relies on the computation of a multivariate potential scale reduction factor (MPSRF) defined as

$$R = \frac{T-1}{T} + \frac{J+1}{J}\lambda\left(\mathbf{V}_{\text{intra}}^{-1}\mathbf{V}_{\text{inter}}\right),$$

where $\lambda(\mathbf{A})$ denotes the largest eigenvalue of matrix $\mathbf{A}$. $\mathbf{V}_{\text{intra}}$ and $\mathbf{V}_{\text{inter}}$ are respectively intra-chain and inter-chain covariance matrices estimated from J independent Markov chains $\{x_{j,t}; j = 1, \cdots, J; t = 1, \cdots, T\}$ of length T:

$$\mathbf{V}_{\text{intra}} = \frac{1}{J(T-1)}\sum_{j,t}\left(x_{j,t} - \overline{x}_j\right)\left(x_{j,t} - \overline{x}_j\right)^{\text{t}},$$

$$\mathbf{V}_{\text{inter}} = \frac{1}{J-1}\sum_{j}\left(\overline{\boldsymbol{x}}_j - \overline{\boldsymbol{x}}\right)\left(\overline{\boldsymbol{x}}_j - \overline{\boldsymbol{x}}\right)^{\text{t}},$$

where $\overline{x}_j$ and $\overline{\boldsymbol{x}}$ denote the empirical mean of chain j and the global mean, respectively. In the following section, we focus on the MPSRFs measured on the amplitude chains. A lower MPSRF value means a better mixing property. Convergence of the chains is diagnosed when the MPSRF is close to one. As suggested in [55], we have chosen a threshold of 1.2.

In order to compute the MPSRF, the chains are divided into batches of Δ_T samples each, and the convergence is detected every Δ_T samples only, so that a difference of $\pm\Delta_T$ is not significant. Moreover, the MPSRF is computed over the second halves of the Markov chains to get rid of the burn-in time period.

For all experiments presented in this paper, we have used $J = 10$ Markov chains. We have also set $\Delta_T = 1000$, and a maximum number of iterations $T_{\text{max}} = 10^5$. The simulations were run using MATLAB on a computer with Intel Xeon Gold 6226R processors, with a CPUs clocked at 2.9 GHz. Each experiment needed 6000 simulated chains, and took four days of computations, including the computation related to the MPSRF. We intentionally performed the computations on a single thread, in order to measure CPU times in a reliable way. This means that in practice, computing times will be significantly smaller than the obtained CPU times, with an acceleration factor depending on the number of available threads. Moreover, since PCGS involves more complex matrix operations than standard Gibbs, multi-threaded computations could be more favorable to PCGS.

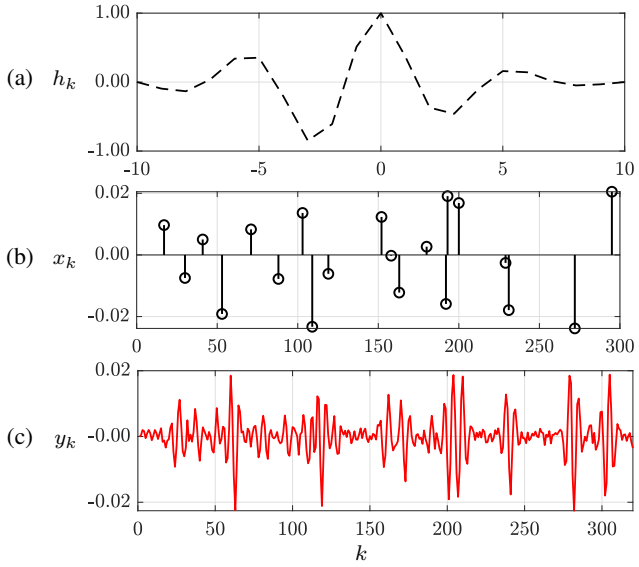

Fig. 2. (a) Impulse response $\boldsymbol{h}$, (b) sample of a BL signal $\boldsymbol{x}$, and (c) corresponding data vector $\boldsymbol{y}$ in the case where $\text{SNR}_{\boldsymbol{y}} = 12\,\text{dB}$.

Once the convergence is diagnosed (or when the maximum number of iterations is reached), the location of the nonzero amplitudes are estimated according to:

$$\hat{q}_k = \begin{cases} 1, & \text{if } \sum_{t\in E} q_k^{(t)} > 0.5|E|, \\ 0, & \text{otherwise,} \end{cases}$$

with E the set of indices t such that $t \in (T_c, T_c + 1000]$, T_c being the index at which the chains have reached the convergence threshold, so that $|E| = 1000$. Then the nonzero amplitudes are estimated by computing the empirical mean of $x_k|q_k = 1$ as follows:

$$\hat{x}_k = \frac{\sum_{t\in E} x_k^{(t)}}{\sum_{t\in E} q_k^{(t)}}.$$

The hyper-parameters are estimated by computing the empirical mean of their respective chains over the set E.

B. Sparse Deconvolution: Bernoulli-Laplace (BL)

A dataset of 300 simulated sparse signals of length $K = 300$ was generated according to a Bernoulli-Laplace (BL) prior. The Laplace density is a well-known case that admits an exact decomposition as a scale mixture of Gaussians distribution. The parameter of the Bernoulli sequence was tuned such that the number of non-zero components ranges between $L \approx 12$ and $L \approx 30$, and the amplitude variance is 0.02. The sparse signals are convolved by an impulse response defined as

$$h_n(f_h) = \cos\left(\frac{n-10}{10}\pi f_h\right) e^{-|0.225n-2|^{1.5}}, \qquad (31)$$

for $n = 0, \ldots, 20$ and $f_h = 3.5$. The dataset is then divided in three, each data vector $\boldsymbol{y}$ being corrupted with a centered i.i.d. Gaussian noise achieving a signal-to-noise ratio $\text{SNR}_{\boldsymbol{y}}$ of 15, 12 and 9 dB, respectively. An example is given in Fig. 2 for $\text{SNR}_{\boldsymbol{y}} = 9\,\text{dB}$.

For each data vector, we have performed a deconvolution with a BL prior, using either the PCGS sampler or a standard Gibbs sampler. It is unsupervised in the sense that hyper-parameters $\boldsymbol{\theta} = [\xi, \sigma_\epsilon, s]$ are unknown. Fig. 3 gives an

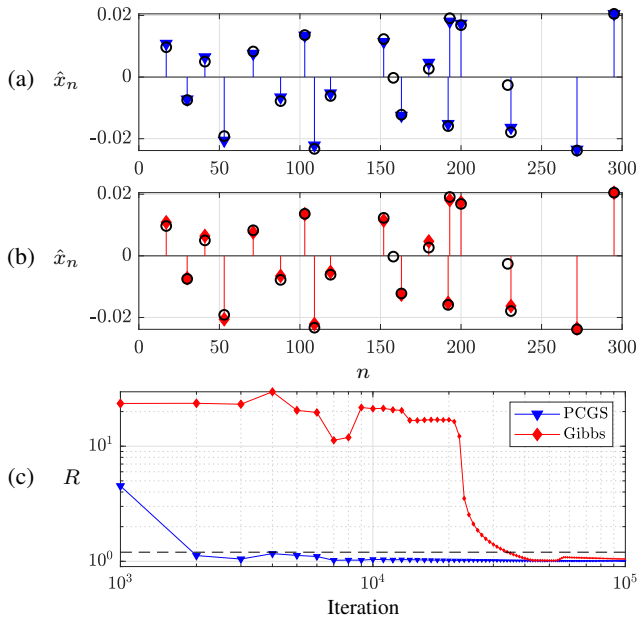

Fig. 3. Example of restored sparse signals at convergence for (a) the PCGS (in blue) and (b) Gibbs (in red), compared to the true values (black circles). (c) Evolution of the MPSRF in log scale. The horizontal dashed line corresponds to the 1.2 threshold.

example in a case where both samplers yield close results. The evolution of the MPSRF w.r.t. the number of iterations is also represented. We can see that the PCGS sampler takes about 2×10^3 iterations to reach convergence, while the Gibbs sampler requires 35×10^3 iterations. The two samplers converge in about 6 and 56 seconds, respectively.

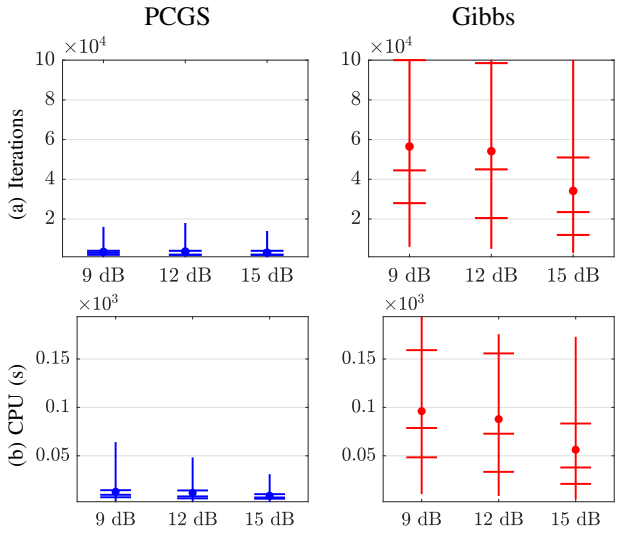

Fig. 4. (a) Number of iterations and (b) CPU time at convergence among 100 trials in the BL case. PCGS in blue, standard Gibbs in red. Results are spread along the vertical lines from the min to the max value. Horizontal segments correspond to divisions into quartiles, and dots indicate the mean value.

Fig. 4 summarizes the results of the whole simulation test using box plots. For each $\mathrm{SNR}_{\boldsymbol{y}}$, we analyze the number of

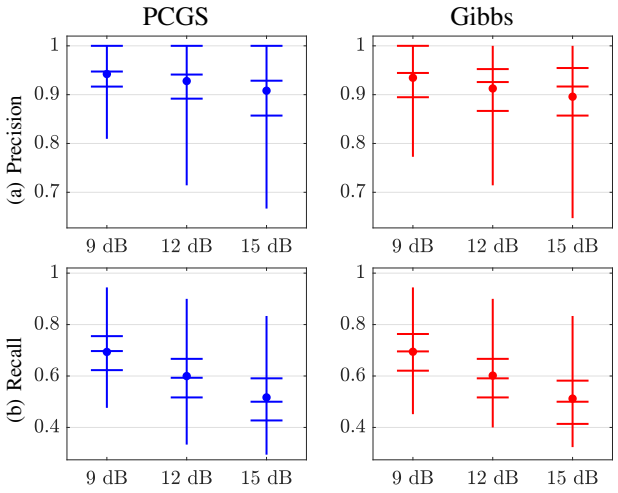

Fig. 5. (a) Precision and (b) recall of the Bernoulli sequences $\boldsymbol{q}$ estimates in the BL case.

iterations and the CPU time required to reach convergence for each sampler among the 100 simulated cases. PCGS clearly needs fewer iterations than standard Gibbs to converge. Moreover, its number of iterations has much smaller variations. These results meet the conclusions of [14], that the PCGS strategy enhances the mixing properties of the sampler, enabling a better exploration of the posterior distribution. PCGS always converged in less than $T_{\max} = 10^5$ iterations (actually, all the chains converged in less than 2×10^4 iterations), whereas the Gibbs sampler failed to converge within $T_{\max}$ iterations in 59 cases, representing about 20% of the 300 datasets.

Although the cost per iteration is larger for PCGS, it needs less CPU time thanks to our recursive implementation. In terms of average CPU time, the two samplers required 11 and 80 seconds respectively, which represents an average acceleration factor of at least[1] 7.

Fig. 5 gives a statistical analysis of *precision* and *recall* [56] of the estimates $\hat{q}_k$. A precision score of one would mean that every detected spike is a true one, even if not all true spikes were detected. A recall score of one would mean that all true spikes were detected, even if irrelevant spikes were also detected. The results show that the two samplers yields similar estimation results, although slightly in favor of PCGS. In particular, according to Fig. 5(a), more than 90% of the detected spikes are relevant on average, at all three $\mathrm{SNR}_{\boldsymbol{y}}$ values. We attribute the lower quality of the Gibbs output to the fact that Gibbs chains are poorly mixing, except at low SNR. The erratic convergence of the Gibbs sampler has already been reported in the BG case [14], [21].

Finally, the average recall values range between 0.5 to 0.7, indicating that a significant proportion of spikes are not detected. However, in the context of noisy sparse deconvolution, spikes with small amplitudes cannot be distinguished from the noise (this is clear on the example of Figs. 2 and 3), which also explains why the recall value decreases with smaller SNR values.

[1] since some Gibbs simulations have not reached convergence.

C. Nonnegative Sparse deconvolution: Bernoulli-Truncated-Gaussian (BTG)

Let us now consider the problem of nonnegative sparse deconvolution when the nonzero signal amplitudes are distributed according to a truncated Gaussian. This is a case where the target distribution does not admit an exact decomposition as a mixture of Gaussian distributions.

A dataset of 300 sparse signals was generated according to a Bernoulli-Truncated-Gaussian (BTG) model [6]. The rest of the numerical procedure is identical to that of Sect. V-B.

The PCGS sampler and a standard Gibbs sampler were run on each of the 300 data vectors. In the former case, we relied on the results of Sect. II-D to approximate the BTG prior. More precisely, we set $s = 1$ and $p_W \sim \mathcal{N}^+(0, \beta^{-2})$, the value of β being a priori fixed to $\beta = 10$, such that $x_k|q_k = 1 \sim NTG(0, 1, 10)$. Let us stress that parameter β controls approximation quality, and more specifically the probability of having negative amplitudes. Here, our choice yields $P(x_k \leq 0|q_k = 1) \approx 0.03$ and $\mathrm{TV} = 0.1$.

In the considered unsupervised framework, hyper-parameters $\boldsymbol{\theta} = [\xi, \sigma_\epsilon, s]$ are unknown. In addition, we consider a semi-blind scenario where the parameter f_h of the impulse response is also unknown. It is thus sampled using a Metropolis-Hasting step similarly to [6].

Fig. 7 gives an example of the estimated sparse signal $\boldsymbol{x}$, where the two samplers yield indistinguishable estimates, despite the approximation needed by PCGS. Fig. 7(c) shows the evolution of the MPSRF w.r.t. the number of iterations. PCGS and Gibbs converged in 2×10^3 and 31×10^3 iterations, respectively. Akin to Fig. 3(c), Fig. 7(c) shows that the MPSRF decreases in a regular way for PCGS, while it is more erratic for standard Gibbs. In this case, the two samplers converged in about 10 and 33 seconds, respectively, which means that PCGS was approximately three times faster than standard Gibbs.

Fig. 8 summarizes the set of results. The computational efficiency of Gibbs sampler still suffers from the same limitations as in the previous experiment. In particular, it produces chains for which the iteration number before convergence strongly varies, which is not the case for PCGS. The Gibbs sampler even failed to pass the MPSRF convergence test within $T_{\max} = 10^5$ iterations in 79 cases (around 26% of the 300 cases), while PCGS always converged in less than 2×10^4

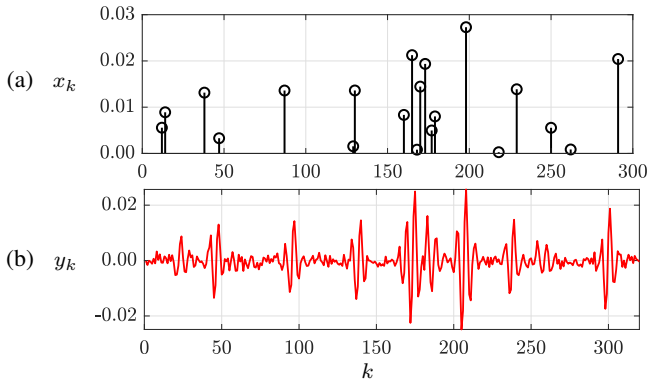

Fig. 6. (a) Sample of a BE signal $\boldsymbol{x}$, and (b) corresponding data vector $\boldsymbol{y}$ in the case where $\mathrm{SNR}_{\boldsymbol{y}} = 12\,\mathrm{dB}$.

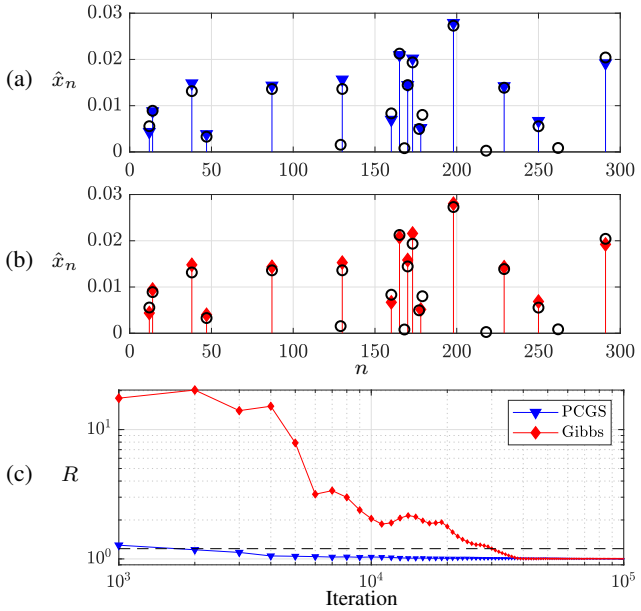

Fig. 7. Example of restored nonnegative sparse signals at convergence for (a) PCGS (in blue) and (b) Gibbs (in red), compared to the true values (black circles). (c) Evolution of the MPSRF in log scale. The horizontal dashed line corresponds to the 1.2 threshold.

iterations. In terms of CPU time, the average acceleration factor is at least 4, as the two samplers require on average 14 and at least 55 seconds. Given the convergence issues of the Gibbs sampler, the true acceleration factor is significantly larger than 4.

In terms of signal restoration quality, Fig. 9 shows that PCGS has similar overall performances than Gibbs. In particular, the recall values are comparable for both samplers. The precision, however, is slightly lower for PCGS in this case, which is due to the approximation required by the former. The precision of the PCGS could be improved by choosing a larger value of parameter β, which controls the quality of the ELSA approximation, at the price of a lower sampling efficiency according to Proposition II.1-(ii).

Moreover, despite the approximation required by PCGS, in particular the fact that the support of the ELSA distribution is not restricted to $\mathbb{R}_+$, none of the 300 estimated signals contains a negative amplitude. This confirms that in the tested examples, the adopted value of β yields a PCGS sampler with both a favorable computational efficiency (compared to the standard Gibbs alternative), and a good approximation of the posterior probability.

VI. Conclusion

In this paper, we introduced a new class of hierarchical prior model for sparse signal restoration, called Bernoulli-Gaussian-Mixture (BGM). It incorporates two levels of Gaussian mixtures:

- The first level corresponds to a two-class mixture, leading to a Bernoulli-$\mathcal{D}$ model, where the Bernoulli variables induce an explicit detection stage in the signal restoration process.

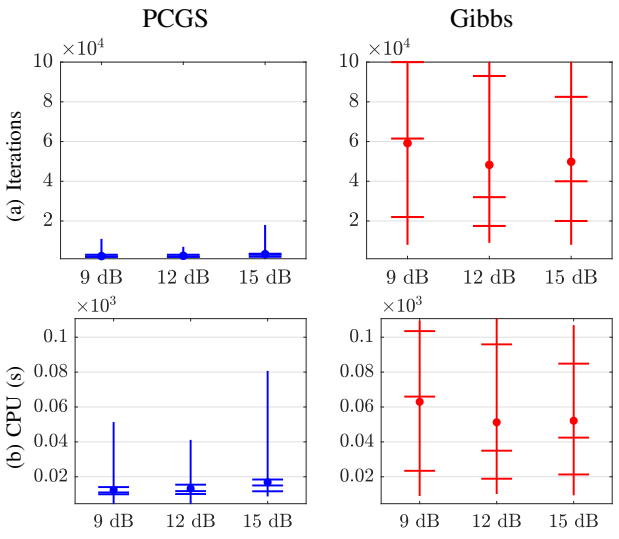

Fig. 8. (a) Number of iterations and (b) CPU time at convergence in the BTG case.

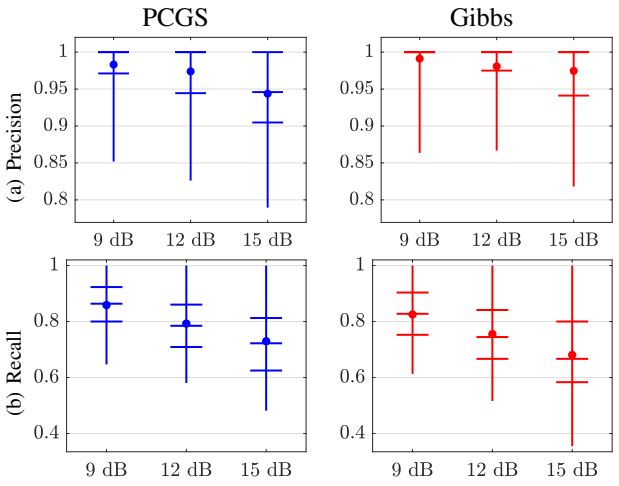

Fig. 9. (a) Precision and (b) recall of the Bernoulli sequences $\boldsymbol{q}$ estimates in the BTG case.

- The second step is a continuous Gaussian mixture decomposition belonging to the class of location-scale mixtures of Gaussians (LSMG).

The LSMG family is rich enough to incorporate scale mixtures of Gaussians, which are symmetric, but also asymmetric densities. In this paper, we paid particular attention to the limit case where $\mathcal{D}$ approximates a distribution restricted to the positive half-line.

In order to efficiently produce samples from the posterior, we proposed an extension of the partially collapsed Gibbs sampling scheme proposed in [14] in the Bernoulli-Gaussian case. Extensive simulation results have shown that the proposed scheme enjoys significantly better mixing properties than the standard Gibbs strategy.

A practical limit of our contribution is found when the number of nonzero components to restore is large, since the cost per iteration of the proposed PCGS scales as $O(L^2)$, where L is the number of nonzero elements at the current iteration. Indeed, PCGS has a computing cost per iteration comparable to that of a deterministic, greedy algorithm such as SBR [57], while it needs a larger number of iterations. The latter point is justified, since the goal of PCGS is to sample from the posterior law, whereas SBR only solves a local minimization problem.

An interesting perspective will be to consider the case of densities supported on an interval, such as the beta density. For such densities, LSMG do not provide approximations of an acceptable quality, and it is an open question to find a suited family of continuous Gaussian mixture models.

Appendix A
Proof of Proposition II.1

(i) Given (9) and $\mu = 0$, it is a simple matter to check that the pdf of $X^* = \beta W$ is p^*. Moreover, according to (8), we have

$$X - X^* = \sqrt{W} Z = \frac{V}{\sqrt{\beta}}, \tag{32}$$

where $V = Z\sqrt{X^*}$. Finally, for any random variable V with pdf p_V, the scaled version (32) converges in probability towards 0 when $\beta \to \infty$, since for any $\epsilon > 0$,

$$P\left(\frac{|V|}{\sqrt{\beta}} \geq \epsilon\right) = 1 - \int_{-\epsilon\sqrt{\beta}}^{\epsilon\sqrt{\beta}} p_V(v)\, dv \to 0.$$

(ii) We have $X = X^* + \sqrt{W} Z$, so it is immediate that $\mathrm{E}[X] = \mathrm{E}[X^*]$ because Z is zero-mean and independent from W. Moreover, Markov's inequality applied to X^* states that

$$P(X^* \geq a) \leq \frac{\mathrm{E}[X^*]}{a}$$

for all $a > 0$. Hence, $\mathrm{E}[X^*] \geq aP(X^* \geq a)$, and there is at least an $a > 0$ for which $P(X^* \geq a) > 0$, otherwise X^* would be almost surely zero (in contradiction with the fact that X^* admits a pdf).

Moreover,

$$\begin{aligned}\mathrm{E}\left[X^2\right] &= \mathrm{E}\left[(X^*)^2\right] + \mathrm{E}\left[WZ^2\right] + 2\mathrm{E}\left[\sqrt{W} X^* Z\right] \\ &= \mathrm{E}\left[(X^*)^2\right] + \sigma_z^2 \mathrm{E}[W]\,,\end{aligned}$$

because Z is independent from $\sqrt{W} X^*$. We obtain (11) since $W = \frac{1}{\beta} X^*$.

Finally, $\mathrm{Corr}(X, W) = \mathrm{Corr}(X, X^*)$ since there is a linear relation between W and X^*. Additionally, $\mathrm{E}[XX^*] = \mathrm{E}\left[(X^*)^2 + \sqrt{W} X^* Z\right] = \mathrm{E}\left[(X^*)^2\right]$, so the Pearson correlation coefficient between X and X^* reads

$$\mathrm{Corr}(X, X^*) = \frac{\mathrm{E}[XX^*] - \mathrm{E}[X]\,\mathrm{E}[X^*]}{\sqrt{\mathrm{var}(X)\mathrm{var}(X^*)}} = \sqrt{\frac{\mathrm{var}(X^*)}{\mathrm{var}(X)}},$$

which leads to (12) after simplification.

Appendix B
Efficient Implementation of PCGS

In Appendix B-A, we show how we can rely on a recursive update of matrix $\mathbf{B}$ to efficiently compute ψ, and thus the ratios $r_{uu'}$.

A. Computation of ψ

Using the matrix inversion lemma, we can deduce from (20) that

$$\mathbf{B} = \mathbf{\Gamma}_\epsilon + \overline{\mathbf{H}}\mathbf{\Gamma}_0\overline{\mathbf{H}}^{\mathrm{t}}. \tag{33}$$

Moreover,

$$\begin{aligned}\mathbf{B}_{(0)} &= \mathbf{\Gamma}_\epsilon + \sum_{\ell\neq k|q_\ell=1} s^2 w_\ell \boldsymbol{h}_\ell \boldsymbol{h}_\ell^{\mathrm{t}},\\ \mathbf{B}_{(1)} &= \mathbf{B}_{(0)} + s^2 w'_k \boldsymbol{h}_k \boldsymbol{h}_k^{\mathrm{t}},\end{aligned}$$

so that

$$\mathbf{B}_{(1-q_k)} - \mathbf{B}_{(q_k)} = (-1)^{q_k} s^2 w'_k \boldsymbol{h}_k \boldsymbol{h}_k^{\mathrm{t}} \tag{34}$$

Using the matrix inversion lemma again yields

$$\begin{aligned}\mathbf{B}_{(1-q_k)}^{-1} &= \left(\mathbf{B}_{(q_k)} + (-1)^{q_k} s^2 w'_k \boldsymbol{h}_k \boldsymbol{h}_k^{\mathrm{t}}\right)^{-1}\\ &= \mathbf{B}_{(q_k)}^{-1} - \mathbf{B}_{(q_k)}^{-1}\boldsymbol{h}_k(\rho'_{q_k})^{-1}\boldsymbol{h}_k^{\mathrm{t}}\mathbf{B}_{(q_k)}^{-1}\end{aligned}$$

where $\rho'_{q_k} = \rho_{q_k}(w'_k) = (-1)^{q_k}(s^2 w'_k)^{-1} + \boldsymbol{h}_k^{\mathrm{t}}\mathbf{B}_{(q_k)}^{-1}\boldsymbol{h}_k$.

Hence,

$$\mathbf{B}_{(1-q_k)}^{-1} - \mathbf{B}_{(q_k)}^{-1} = -(\rho'_{q_k})^{-1}\mathbf{B}_{(q_k)}^{-1}\boldsymbol{h}_k\boldsymbol{h}_k^{\mathrm{t}}\mathbf{B}_{(q_k)}^{-1}$$

so

$$\boldsymbol{y}^{\mathrm{t}}\left(\mathbf{B}_{(1-q_k)}^{-1} - \mathbf{B}_{(q_k)}^{-1}\right)\boldsymbol{y} = (\rho'_{q_k})^{-1}(\gamma'_{q_k})^2 \tag{35}$$

with $\gamma'_{q_k} = \gamma_{q_k}(w'_k) = \boldsymbol{h}_k^{\mathrm{t}}\mathbf{B}_{(q_k)}^{-1}\boldsymbol{y}$.

Taking the determinant of (34) using the matrix determinant lemma,

$$\begin{aligned}\left|\mathbf{B}_{(1-q_k)}\right| &= \left|\mathbf{B}_{(q_k)} + (-1)^{q_k} s^2 w'_k \boldsymbol{h}_k \boldsymbol{h}_k^{\mathrm{t}}\right|\\ &= \left(1 + \boldsymbol{h}_k^{\mathrm{t}}\mathbf{B}_{(q_k)}^{-1}(-1)^{q_k} s^2 w'_k \boldsymbol{h}_k\right)\ \left|\mathbf{B}_{(q_k)}\right|\\ \left|\mathbf{B}_{(1-q_k)}\right| &= (-1)^{q_k} s^2 w'_k \rho'_{q_k}\left|\mathbf{B}_{(q_k)}\right|.\end{aligned} \tag{36}$$

From (35) and (36) we have

$$\psi(\mathbf{B}_{(1-q_k)}, \mathbf{B}_{(q_k)}) = \ln\left((-1)^{q_k} s^2 w'_k \rho'_{q_k}\right) - \frac{(\gamma'_{q_k})^2}{\rho'_{q_k}} \tag{37}$$

According to (23) and (24), (37) is applicable to the computation of r_{01} (with $q_k = 0$) and of r_{10} (with $q_k = 1$ and $w'_k = w_k$). For the update step, one can use that:

$$\begin{aligned}\psi(\mathbf{B}_1, \mathbf{B}) &= \psi(\mathbf{B}_1, \mathbf{B}_0) - \psi(\mathbf{B}, \mathbf{B}_0),\\ &= \ln\left(\frac{w'_k\rho'_0}{w_k\rho_0}\right) - \frac{(\gamma'_0)^2}{\rho'_0} + \frac{\gamma_0^2}{\rho_0}.\end{aligned} \tag{38}$$

We propose to rely on (37) and (38) to evaluate function ψ, but we still need an efficient way to compute ρ_{q_k} and γ_{q_k}. One possibility is to rely on the matrix inversion lemma again. Akin to [14], we rather make use of Cholesky factorizations.

B. Cholesky Factorization

Let $\mathbf{G} = \overline{\mathbf{H}}\sqrt{\mathbf{\Gamma}_0} = s\overline{\mathbf{H}}\sqrt{\mathrm{Diag}\{\boldsymbol{w}\}}$, so that $\mathbf{B} = \mathbf{\Gamma}_\epsilon + \mathbf{G}\mathbf{G}^{\mathrm{t}}$. Using the matrix inversion lemma, we have

$$\mathbf{B}^{-1} = (\mathbf{\Gamma}_\epsilon + \mathbf{G}\mathbf{G}^{\mathrm{t}})^{-1} = \mathbf{\Gamma}_\epsilon^{-1} - \mathbf{\Gamma}_\epsilon^{-1}\mathbf{G}\mathbf{S}^{-1}\mathbf{G}^{\mathrm{t}}\mathbf{\Gamma}_\epsilon^{-1}$$

where

$$\mathbf{S} = \mathbf{G}^{\mathrm{t}}\mathbf{\Gamma}_\epsilon^{-1}\mathbf{G} + \mathbf{I}_L \tag{39}$$

is an $L\times L$ positive-definite matrix. The Cholesky factorization of its inverse reads

$$\mathbf{S}^{-1} = \mathbf{F}^{\mathrm{t}}\mathbf{F}, \tag{40}$$

where $\mathbf{F}$ is an $L \times L$ upper triangular matrix. Akin to [14], we will handle matrix $\mathbf{F}$ rather than $\mathbf{B}$, which is $N \times N$.

Let us stress that ρ'_{q_k} and γ'_{q_k} take the form:

$$\begin{aligned}\rho'_{q_k} &= (-1)^{q_k}(s^2 w'_k)^{-1} + \boldsymbol{h}_k^{\mathrm{t}}\mathbf{\Gamma}_\epsilon^{-1}\boldsymbol{h}_k\\ &\quad - \left(\mathbf{F}_{(q_k)}\mathbf{G}_{(q_k)}^{\mathrm{t}}\mathbf{\Gamma}_\epsilon^{-1}\boldsymbol{h}_k\right)^{\mathrm{t}}\left(\mathbf{F}_{(q_k)}\mathbf{G}_{(q_k)}^{\mathrm{t}}\mathbf{\Gamma}_\epsilon^{-1}\boldsymbol{h}_k\right),\end{aligned} \tag{41}$$

$$\gamma'_{q_k} = \boldsymbol{h}_k^{\mathrm{t}}\mathbf{\Gamma}_\epsilon^{-1}\boldsymbol{y} - \left(\mathbf{F}_{(q_k)}\mathbf{G}_{(q_k)}^{\mathrm{t}}\mathbf{\Gamma}_\epsilon^{-1}\boldsymbol{h}_k\right)^{\mathrm{t}}\mathbf{F}_{(q_k)}\mathbf{G}_{(q_k)}^{\mathrm{t}}\mathbf{\Gamma}_\epsilon^{-1}\boldsymbol{y}. \tag{42}$$

C. Efficient Update of matrix $\mathbf{G}$ and Cholesky Factor $\mathbf{F}$

1) Birth moves: In the case of birth moves, the update of matrix $\mathbf{G}$ is straightforward if the new column $s\sqrt{w_k}\boldsymbol{h}_k$ is simply added at the end of the matrix. Let $\mathbf{G}_k$ denotes the matrix $\mathbf{G}_{(1)}$ for which the column $s\sqrt{w_k}\boldsymbol{h}_k$ is the last one, *i.e.*, $\mathbf{G}_k = \begin{bmatrix}\mathbf{G}_{(0)} & s\sqrt{w_k}\boldsymbol{h}_k\end{bmatrix}$.

Note that the order in which the columns are stacked in matrix $\mathbf{G}$ has a direct influence on matrix $\mathbf{F}$. To keep track of the column positioning, we introduce an index vector $\boldsymbol{o} = [o_1, \cdots, o_L]$ such that o_k is the position of column $s\sqrt{w_k}\boldsymbol{h}_k$ inside matrix $\mathbf{G}$. Let $\mathbf{S}_k = \mathbf{S}_{(1)}$ and $\mathbf{F}_k = \mathbf{F}_{(1)}$ when $\mathbf{G}_k = \mathbf{G}_{(0)}$ with

$$\mathbf{F}_k = \begin{bmatrix}\mathbf{F}_{11} & \boldsymbol{f}_{12}\\ \mathbf{0} & f_{22}\end{bmatrix}, \tag{43}$$

where $\mathbf{F}_{11}$, $\boldsymbol{f}_{12}$ and f_{22} are an upper triangular matrix, a column vector and a scalar, respectively. It follows that

$$\mathbf{F}_k^{\mathrm{t}}\mathbf{F}_k = \begin{bmatrix}\mathbf{F}_{11}^{\mathrm{t}}\mathbf{F}_{11} & \mathbf{F}_{11}^{\mathrm{t}}\boldsymbol{f}_{12}\\ \boldsymbol{f}_{12}^{\mathrm{t}}\mathbf{F}_{11} & \boldsymbol{f}_{12}^{\mathrm{t}}\boldsymbol{f}_{12} + f_{22}^2\end{bmatrix} \tag{44}$$

For a birth move, given the definition of $\mathbf{G}_k$, by (39), (40) and using a block-wise matrix inversion formula, the update from $\mathbf{F}_{(0)}$ to $\mathbf{F}_{(1)}$ yields

$$\mathbf{F}_k^{\mathrm{t}}\mathbf{F}_k = \mathbf{F}_{(1)}^{\mathrm{t}}\mathbf{F}_{(1)} = \begin{bmatrix}\mathbf{F}_{(0)}^{\mathrm{t}}\mathbf{F}_{(0)} + \rho_0\boldsymbol{b}_0\boldsymbol{b}_0^{\mathrm{t}} & s^{-1}w_k^{-\frac{1}{2}}\boldsymbol{b}_0\\ s^{-1}w_k^{-\frac{1}{2}}\boldsymbol{b}_0^{\mathrm{t}} & (s^2 w_k)^{-1}\rho_0^{-1}\end{bmatrix} \tag{45}$$

where $\boldsymbol{b}_0 = -\mathbf{F}_{(0)}^{\mathrm{t}}\mathbf{F}_{(0)}^{-1}\mathbf{G}_{(0)}^{\mathrm{t}}\mathbf{\Gamma}_\epsilon^{-1}\boldsymbol{h}_k\rho_0^{-1}$. It is straightforward to see that $\mathbf{F}_{(1)}^{\mathrm{t}}\mathbf{F}_{(1)} = \overline{\mathbf{F}}_{(0)}^{\mathrm{t}}\overline{\mathbf{F}}_{(0)} + \boldsymbol{v}\boldsymbol{v}^{\mathrm{t}}$ with

$$\overline{\mathbf{F}}_{(0)} = \begin{bmatrix}\mathbf{F}_{(0)} & \mathbf{0}^{\mathrm{t}}\\ \mathbf{0} & 0\end{bmatrix} \text{ and } \boldsymbol{v} = \begin{bmatrix}\boldsymbol{v}_1\\ \boldsymbol{v}_2\end{bmatrix} = \begin{bmatrix}\boldsymbol{b}_0\sqrt{\rho_0}\\ s^{-1}w_k^{-\frac{1}{2}}/\sqrt{\rho_0}\end{bmatrix}. \tag{46}$$

As a consequence, $\mathbf{F}_{(1)}$ can be computed using a rank-1 Cholesky update:

$$\mathbf{F}_{(1)} = \mathrm{cholupdate}(\overline{\mathbf{F}}_{(0)}, \boldsymbol{v}, +). \tag{47}$$

2) Death moves: For a death move, $s\sqrt{w_k}\boldsymbol{h}_k$ must be removed and vector $\boldsymbol{o}$ must be updated accordingly.

When the column to be removed is the last one, *i.e.*, $o_k = L$, the updating is straightforward. From (45) we have:

$$\begin{aligned}\mathbf{F}_{(0)}^{\mathrm{t}}\mathbf{F}_{(0)} &= \mathbf{F}_{11}^{\mathrm{t}}\mathbf{F}_{11} - \rho_0\boldsymbol{b}_0\boldsymbol{b}_0^{\mathrm{t}}\\ &= \mathbf{F}_{11}^{\mathrm{t}}\mathbf{F}_{11} - \boldsymbol{v}_1\boldsymbol{v}_1^{\mathrm{t}}\end{aligned}$$

thus $\mathbf{F}_{(0)}$ can be computed using a rank-1 Cholesky downdate:

$$\mathbf{F}_{(0)} = \text{cholupdate}(\mathbf{F}_{11}, \boldsymbol{v}_1, -) \tag{48}$$

where $\boldsymbol{v}_1 = \boldsymbol{b}_0\sqrt{\rho_0}$. In this case, ρ_0 and $\boldsymbol{b}_0$ can be directly extracted from matrix $\mathbf{F}_{(1)}$ by identifying (44) with (45):

$$\begin{aligned}\rho_0 &= \left(s^2 w_k(\boldsymbol{f}_{12}^{\mathrm{t}}\boldsymbol{f}_{12} + f_{22}^2)\right)^{-1},\\ \boldsymbol{b}_0 &= s\sqrt{w_k}\,\mathbf{F}_{11}^{\mathrm{t}}\boldsymbol{f}_{12}.\end{aligned}$$

When $o_k < L$, *i.e.*, $\mathbf{G}_{(1)} = [\mathbf{G}_\ell \quad s\sqrt{w_k}\boldsymbol{h}_k \quad \mathbf{G}_r]$, a permutation matrix

$$\mathbf{P} = \begin{bmatrix}\mathbf{I}_\ell & \mathbf{0} & \mathbf{0}\\ 0 & 0 & 1\\ \mathbf{0} & \mathbf{I}_r & \mathbf{0}\end{bmatrix}$$

needs to be introduced, such that: $\mathbf{G}_{(1)}\mathbf{P} = [\mathbf{G}_\ell \quad \mathbf{G}_r \quad s\sqrt{w_k}\boldsymbol{h}_k] = \mathbf{G}_k$ and

$$\mathbf{F}_k^{\mathrm{t}}\mathbf{F}_k = (\mathbf{P}^{\mathrm{t}}\mathbf{F}_{(1)}\mathbf{P})^{\mathrm{t}}(\mathbf{P}^{\mathrm{t}}\mathbf{F}_{(1)}\mathbf{P}). \tag{49}$$

However, one has to keep in mind that $\mathbf{F}_k^{\mathrm{t}} \neq \mathbf{P}^{\mathrm{t}}\mathbf{F}_{(1)}\mathbf{P}$ given that $\mathbf{P}^{\mathrm{t}}\mathbf{F}_{(1)}\mathbf{P}$ is not a triangular matrix. Nevertheless, (49) allows to compute $\boldsymbol{b}_0$ et ρ_0 directly from $\mathbf{F}_{(1)}$ and $\mathbf{F}_{11}$ using an additional rank-1 Cholesky update. Let

$$\mathbf{F}_{(1)} = \begin{bmatrix}\mathbf{f}_l & \mathbf{f}_k & \mathbf{f}_r\end{bmatrix} = \begin{bmatrix}\mathbf{f}_{ll} & \mathbf{f}_{lk} & \mathbf{f}_{lr}\\ \mathbf{0} & \mathbf{f}_{kk} & \mathbf{f}_{kr}\\ \mathbf{0} & \mathbf{0} & \mathbf{f}_{rr}\end{bmatrix} \tag{50}$$

From (44), (45), (49) and (50) we have

$$\rho_0^{-1} = s^2 w_k \mathbf{f}_k^2, \quad \text{and} \quad \boldsymbol{b}_0 = s\sqrt{w_k}[\mathbf{f}_l \quad \mathbf{f}_r]^{\mathrm{t}}\mathbf{f}_m,$$

and $\mathbf{F}_{11} = \begin{bmatrix}\mathbf{f}_{ll} & \mathbf{f}_{lr}\\ \mathbf{0} & \mathbf{f}_*\end{bmatrix}$ with $\mathbf{f}_* = \text{cholupdate}(\mathbf{f}_{rr}, \mathbf{f}_{kr}, +)$. Finally, $\mathbf{F}_{(0)}$ can be computed via (48).

3) Update moves: In the case of an update move, matrices $\mathbf{G}$ and $\mathbf{F}$ keep the same size. For matrix $\mathbf{G}$, one has to substitute $s\sqrt{w_k}\boldsymbol{h}_k$ with $s\sqrt{w_k'}\boldsymbol{h}_k$. For matrix $\mathbf{F}$, one has to perform a death update of column $s\sqrt{w_k}\boldsymbol{h}_k$ followed by a birth update of $s\sqrt{w_k'}\boldsymbol{h}_k$.

D. Computation of ϕ

Recall that ϕ_u is defined by (30). Its computation can be simplified using the results of Appendix B-C. From (40), (47) and (48) one can show that:

$$\phi_1 - \phi = \beta^2\left(w_k' - V^2\right) - 2\beta V \boldsymbol{y}^{\mathrm{t}}\boldsymbol{\Gamma}_\epsilon^{-1}\mathbf{G}_{(1)}\boldsymbol{v} \tag{51}$$

in the birth move (23), with $\boldsymbol{v} = \left[\boldsymbol{b}_0\rho_0^{\frac{1}{2}}, (w_k')^{-1}\rho_0^{-\frac{1}{2}}\right]^{\mathrm{t}}$ and $V = \sum_l \sqrt{w_l}v_l$. For the death move (24), $\phi_0 - \phi$ takes the opposite expression of (51) with $w_k' = w_k$.

E. Efficient Sampling of the Amplitudes $\boldsymbol{x}$

From (19) and (39), we have that

$$\begin{aligned}\boldsymbol{\Gamma}_1^{-1} &= \overline{\mathbf{H}}^{\mathrm{t}}\boldsymbol{\Gamma}_\epsilon^{-1}\overline{\mathbf{H}} + \boldsymbol{\Gamma}_0^{-1},\\ &= \boldsymbol{\Gamma}_0^{-\frac{1}{2}}\left(\boldsymbol{\Gamma}_0^{\frac{1}{2}}\overline{\mathbf{H}}^{\mathrm{t}}\boldsymbol{\Gamma}_\epsilon^{-1}\overline{\mathbf{H}}\boldsymbol{\Gamma}_0^{\frac{1}{2}} + \mathbf{I}_N\right)\boldsymbol{\Gamma}_0^{-\frac{1}{2}},\\ &= \boldsymbol{\Gamma}_0^{-\frac{1}{2}}\mathbf{S}\boldsymbol{\Gamma}_0^{-\frac{1}{2}},\end{aligned}$$

so $\boldsymbol{\Gamma}_1 = \boldsymbol{\Gamma}_0^{\frac{1}{2}}\mathbf{S}^{-1}\boldsymbol{\Gamma}_0^{\frac{1}{2}} = \boldsymbol{\Gamma}_0^{\frac{1}{2}}\mathbf{F}^{\mathrm{t}}\mathbf{F}\boldsymbol{\Gamma}_0^{\frac{1}{2}}$, and hence,

$$\boldsymbol{\mu}_1 = \boldsymbol{\Gamma}_0^{\frac{1}{2}}\mathbf{F}^{\mathrm{t}}\left(\mathbf{F}\mathbf{G}^{\mathrm{t}}\boldsymbol{\Gamma}_\epsilon^{-1}\boldsymbol{y} + \mathbf{F}\boldsymbol{\Gamma}_0^{-\frac{1}{2}}\boldsymbol{\mu}_x\right).$$

Finally, the amplitudes $\boldsymbol{x}$ can be sampled efficiently as $\overline{\boldsymbol{x}} = \boldsymbol{\mu}_1 + \boldsymbol{\Gamma}_0^{\frac{1}{2}}\mathbf{F}^{\mathrm{t}}\boldsymbol{u}$ where $\boldsymbol{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_L)$.

F. Further simplification for problems with a fixed dictionary

The efficient implementation described above can be further optimized when the noise is considered i.i.d. and the quantities $\widetilde{\mathbf{H}} = \mathbf{H}^{\mathrm{t}}\mathbf{H}$ and $\boldsymbol{z} = \mathbf{H}^{\mathrm{t}}\boldsymbol{y}$ can be computed and stored beforehand. The strategy is to iteratively update $\widetilde{\mathbf{G}} = \mathbf{G}^t\boldsymbol{h}_k$ and $\widetilde{\boldsymbol{z}} = \mathbf{G}^{\mathrm{t}}\boldsymbol{y}$ instead of matrix $\mathbf{G}$. Thus, for a birth move, $\widetilde{\mathbf{G}}$ and $\widetilde{\boldsymbol{z}}$ can be updated as follows:

$$\widetilde{\mathbf{G}}_{(1)} = \left[\widetilde{\mathbf{G}}_{(1)}, \quad s\sqrt{w_k}\widetilde{\boldsymbol{h}}_k\right]^{\mathrm{t}}, \ \widetilde{\boldsymbol{z}}_{(1)} = \left[\widetilde{\boldsymbol{z}}_{(0)}, \quad s\sqrt{w_k}z_k\right]^{\mathrm{t}}.$$

For a death move, the column $s\sqrt{w_k}\widetilde{\boldsymbol{h}}_k$ and the scalar $s\sqrt{w_k}z_k$ must be removed from $\widetilde{\mathbf{G}}_{(1)}$ and $\widetilde{\boldsymbol{z}}_{(1)}$, respectively. Finally, for an update move, one has to perform a death move of the current atom followed by a birth move of the new atom.

Given $\widetilde{\mathbf{G}}$ and $\tilde{\boldsymbol{z}}$, the expression of ρ_{q_k}' and γ_{q_k}' yields:

$$\begin{aligned}\rho_{q_k}' &= (-1)^{q_k}(s^2 w_k')^{-1} + \widetilde{h}_{kk}\sigma_\epsilon^{-2},\\ &\quad - \sigma_\epsilon^{-4}\left(\mathbf{F}_{(q_k)}\widetilde{\boldsymbol{g}}_k\right)^{\mathrm{t}}\left(\mathbf{F}_{(q_k)}\widetilde{\boldsymbol{g}}_k\right), \qquad (52)\\ \gamma_{q_k}' &= \widetilde{z}_k\sigma_\epsilon^{-2} - \sigma_\epsilon^{-4}\left(\mathbf{F}_{(q_k)}\widetilde{\boldsymbol{g}}_k\right)^{\mathrm{t}}\mathbf{F}_{(q_k)}\tilde{\boldsymbol{z}}. \qquad (53)\end{aligned}$$

Note that the same strategy can be used when the dictionary is sparse, even in the case where $\widetilde{\mathbf{H}} = \mathbf{H}^{\mathrm{t}}\mathbf{H}$ and $\boldsymbol{z} = \mathbf{H}^{\mathrm{t}}\boldsymbol{y}$ are computed on-the-fly.

REFERENCES

[1] B. K. Natarajan, "Sparse approximate solutions to linear systems", *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227–234, Apr. 1995.

[2] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification", *International Journal of Control*, vol. 50, no. 5, pp. 1873–1896, May 1989.

[3] D. L. Donoho and Y. Tsaig, "Fast solution of ℓ_1-norm minimization problems when the solution may be sparse", *IEEE Transactions on Information Theory*, vol. 54, no. 11, pp. 4789–4812, Nov. 2008.

[4] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition", in *Asilomar Conference on Signals, Systems and Computers*, Nov. 1993, vol. 1, pp. 40–44.

[5] S. Gulam Razul, W. Fitzgerald, and C. Andrieu, "Bayesian model selection and parameter estimation of nuclear emission spectra using RJMCMC", *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 497, no. 2, pp. 492–510, Feb. 2003.

[6] V. Mazet, J. Idier, and D. Brie, "Déconvolution impulsionnelle positive myope", in *20° Colloque sur le traitement du signal et des images*. GRETSI, Sept. 2005.

[7] I. Barbu and C. Herzet, "A new approach for volume reconstruction in TomoPIV with the alternating direction method of multipliers", *Measurement Science and Technology*, vol. 27, no. 10, pp. 104002, Sept. 2016.

[8] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression", *Annals of Statistics*, vol. 32, no. 2, pp. 407–499, Apr. 2004.

[9] M. Yaghoobi and M. E. Davies, "Fast non-negative orthogonal least squares", in *European Signal Processing Conference (EUSIPCO)*, Aug. 2015, pp. 479–483.

[10] T. T. Nguyen, J. Idier, C. Soussen, and E.-H. Djermoune, "Non-negative orthogonal greedy algorithms", *IEEE Transactions on Signal Processing*, vol. 67, no. 21, pp. 5643–5658, Nov. 2019.

[11] J. J. Kormylo and J. M. Mendel, "Maximum likelihood detection and estimation of Bernoulli-Gaussian processes", *IEEE Transactions on Information Theory*, vol. 28, no. 3, pp. 482–488, May 1982.

[12] F. Champagnat, Y. Goussard, and J. Idier, "Unsupervised deconvolution of sparse spike trains using stochastic approximation", *IEEE Transactions on Signal Processing*, vol. 44, no. 12, pp. 2988–2998, Dec. 1996.

[13] Q. Cheng, R. Chen, and T.-H. Li, "Simultaneous wavelet estimation and deconvolution of reflection seismic signals", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 34, no. 2, pp. 377–384, Mar. 1996.

[14] 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, Apr. 2011.

[15] F. Champagnat and J. Idier, "Deconvolution of sparse spike trains accounting for wavelet phase shifts and colored noise", in *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, Minneapolis, MN, USA, Apr. 1993, vol. III, pp. 452–455.

[16] S. Chaabene, L. Chaari, and A. Kallel, "Sparse Bayesian pMRI reconstruction with complex bernoulli-laplace mixture priors", in *IEEE Middle East Conference on Biomedical Engineering (MECBME)*, Mar. 2018, pp. 193–197.

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

[18] F. Costa, H. Batatia, L. Chaari, and J.-Y. Tourneret, "Sparse EEG source localization using Bernoulli-Laplacian priors", *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 12, pp. 2888–2898, Dec. 2015.

[19] N. Dobigeon, A. O. Hero, and J.-Y. Tourneret, "Hierarchical Bayesian sparse image reconstruction with application to MRFM", *IEEE Transactions on Image Processing*, vol. 18, no. 9, pp. 2059–2070, Sept. 2009.

[20] S. Bourguignon, C. Soussen, H. Carfantan, and J. Idier, "Sparse deconvolution: Comparison of statistical and deterministic approaches", in *IEEE Statistical Signal Processing Workshop (SSP)*, June 2011, pp. 317–320.

[21] S. Bourguignon and H. Carfantan, "Bernoulli-Gaussian spectral analysis of unevenly spaced astrophysical data", in *IEEE Statistical Signal Processing Workshop (SSP)*, July 2005, pp. 811–816.

[22] D. Ge, J. Idier, and E. Le Carpentier, "A new MCMC algorithm for blind Bernoulli-Gaussian deconvolution", in *European Signal Processing Conference (EUSIPCO)*, Aug. 2008, pp. 1–5.

[23] D. A. van Dyk and T. Park, "Partially collapsed Gibbs samplers", *Journal of the American Statistical Association*, vol. 103, no. 482, pp. 790–796, June 2008.

[24] G. Kail, J.-Y. Tourneret, F. Hlawatsch, and N. Dobigeon, "Blind deconvolution of sparse pulse sequences under a minimum distance constraint: A partially collapsed Gibbs sampler method", *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 2727–2743, June 2012.

[25] M. Boudineau, H. Carfantan, S. Bourguignon, and M. Bazot, "Sampling schemes and parameter estimation for nonlinear Bernoulli-Gaussian sparse models", in *IEEE Statistical Signal Processing Workshop (SSP)*, June 2016, pp. 1–5.

[26] F. Champagnat and J. Idier, "A connection between half-quadratic criteria and EM algorithms", *IEEE Signal Processing Letters*, vol. 11, no. 9, pp. 709–712, Sept. 2004.

[27] P. J. Huber, *Robust Statistics*, John Wiley, New York, NY, 1981.

[28] D. Geman and G. Reynolds, "Constrained restoration and the recovery of discontinuities", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 3, pp. 367–383, Mar. 1992.

[29] D. Geman and C. Yang, "Nonlinear image recovery with half-quadratic regularization", *IEEE Transactions on Image Processing*, vol. 4, no. 7, pp. 932–946, July 1995.

[30] 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.

[31] M. West, "On scale mixtures of normal distribution", *Biometrika*, vol. 74, no. 3, pp. 646–648, Sept. 1987.

[32] T. Park and G. Casella, "The Bayesian Lasso", *Journal of the American Statistical Association*, vol. 103, no. 482, pp. 681–686, 2008.

[33] C. Févotte 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, Nov. 2006.

[34] Y. Marnissi, E. Chouzenoux, A. Benazza-Benyahia, and J.-C. Pesquet, "An auxiliary variable method for Markov Chain Monte Carlo algorithms in high dimension", *Entropy*, vol. 20, no. 2, pp. 110, Feb. 2018.

[35] M. Vono, N. Dobigeon, and P. Chainais, "Split-and-augmented Gibbs sampler—application to large-scale inference problems", *IEEE Transactions on Signal Processing*, vol. 67, no. 6, pp. 1648–1661, Mar. 2019.

[36] M. Vono, N. Dobigeon, and P. Chainais, "Asymptotically exact data augmentation: Models, properties, and algorithms", *Journal of Computational and Graphical Statistics*, vol. 30, no. 2, pp. 335–348, Nov. 2020.

[37] H. Snoussi and J. Idier, "Bayesian blind separation of generalized hyperbolic processes in noisy and underdeterminate mixtures", *IEEE Transactions on Signal Processing*, vol. 54, no. 9, pp. 3257–3269, Sept. 2006.

[38] M. C. Amrouche, H. Carfantan, and J. Idier, "A partially collapsed Gibbs sampler for unsupervised nonnegative sparse signal restoration", in *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, Toronto, Canada, June 2021.

[39] O. Barndorff-Nielsen, J. Kent, and M. Sørensen, "Normal variance-mean mixtures and z distributions", *International Statistical Review*, vol. 50, no. 2, pp. 145–159, Aug. 1982.

[40] Y. Yu, "On normal variance–mean mixtures", *Statistics and Probability Letters*, vol. 121, pp. 45–50, Feb. 2017.

[41] E. Eberlein and U. Keller, "Hyperbolic distributions in finance", *Bernoulli*, vol. 1, no. 3, pp. 281–299, Sept. 1995.

[42] R. S. Protassov, "EM-based maximum likelihood parameter estimation for multivariate generalized hyperbolic distributions with fixed λ", *Statistics and Computing*, vol. 14, no. 1, pp. 67–77, Jan. 2004.

[43] D. Karlis and A. Santourian, "Model-based clustering with non-elliptically contoured distributions", *Statistics and Computing*, vol. 19, no. 1, pp. 73–83, Mar. 2009.

[44] D. Wraith and F. Forbes, "Location and scale mixtures of Gaussians with flexible tail behaviour: Properties, inference and application to multivariate clustering", *Computational Statistics and Data Analysis*, vol. 90, pp. 61–73, Oct. 2015.

[45] O. Barndorff-Nielsen and D. G. Kendall, "Exponentially decreasing distributions for the logarithm of particle size", *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 353, no. 1674, pp. 401–419, Mar. 1977.

[46] E. Eberlein and E. A. v. Hammerstein, "Generalized hyperbolic and inverse Gaussian distributions: Limiting cases and approximation of processes", in *Seminar on Stochastic Analysis, Random Fields and Applications IV*, R. C. Dalang, M. Dozzi, and F. Russo, Eds., Basel, 2004, pp. 221–264, Birkhäuser Basel.

[47] S. Kotz, T. Kozubowski, and K. Podgorski, *The Laplace distribution and generalizations*, Springer Science & Business Media, 2001.

[48] B. Jørgensen, *Statistical Properties of the Generalized Inverse Gaussian Distribution*, Lecture Notes in Statistics. Springer-Verlag, New York, 1982.

[49] N. Chopin, "Fast simulation of truncated Gaussian distributions", *Statistics and Computing*, vol. 21, no. 2, pp. 275–288, Jan. 2010.

[50] P. J. Green, "Reversible jump Markov Chain Monte Carlo computation and Bayesian model determination", *Biometrika*, vol. 82, no. 4, pp. 711–732, Dec. 1995.

[51] S. Bourguignon and H. Carfantan, "Spectral analysis of irregularly sampled data using a Bernoulli-Gaussian model with free frequencies", in *IEEE International Conference on Acoustics Speech and Signal Processing Proceedings (ICASSP)*, May 2006, vol. 3, pp. 516–519.

[52] C. Andrieu and A. Doucet, "Joint Bayesian model selection and estimation of noisy sinusoids via reversible jump MCMC", *IEEE Transactions on Signal Processing*, vol. 47, no. 10, pp. 2667–2676, Oct. 1999.

[53] G. O. Roberts and J. S. Rosenthal, "Optimal scaling for various Metropolis-Hastings algorithms", *Statistical Science*, vol. 16, no. 4, pp. 351–367, Nov. 2001.

[54] R. Waagepetersen and D. Sorensen, "A tutorial on reversible jump MCMC with a view toward applications in QTL-mapping", *International Statistical Review*, vol. 69, no. 1, pp. 49–61, May 2001.

[55] S. P. Brooks and A. Gelman, "General methods for monitoring convergence of iterative simulations", *Journal of Computational and Graphical Statistics*, vol. 7, no. 4, pp. 434–455, 1998.

[56] K. M. Ting, "Precision and recall", in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds., pp. 781–781. Springer US, Boston, MA, 2010.

[57] C. Soussen, J. Idier, D. Brie, and J. Duan, "From Bernoulli-Gaussian deconvolution to sparse signal restoration", *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4572–4584, Oct. 2011.