Covariance-Free Sparse Bayesian Learning

Alexander Lin, *Student Member, IEEE*, Andrew H. Song, *Student Member, IEEE*, Berkin Bilgic, and Demba Ba, *Member, IEEE*

***Abstract*—Sparse Bayesian learning (SBL) is a powerful framework for tackling the sparse coding problem while also providing uncertainty quantification. The most popular inference algorithms for SBL exhibit prohibitively large computational costs for high-dimensional problems due to the need to maintain a large covariance matrix. To resolve this issue, we introduce a new method for accelerating SBL inference – named covariance-free expectation maximization (CoFEM) – that avoids explicit computation of the covariance matrix. CoFEM solves multiple linear systems to obtain unbiased estimates of the posterior statistics needed by SBL. This is accomplished by exploiting innovations from numerical linear algebra such as preconditioned conjugate gradient and a little-known diagonal estimation rule. For a large class of compressed sensing matrices, we provide theoretical justifications for why our method scales well in high-dimensional settings. Through simulations, we show that CoFEM can be up to thousands of times faster than existing baselines without sacrificing coding accuracy. Through applications to calcium imaging deconvolution and multi-contrast MRI reconstruction, we show that CoFEM enables SBL to tractably tackle high-dimensional sparse coding problems of practical interest.**

***Index Terms*—Sparse Bayesian learning (SBL), compressed sensing, expectation-maximization algorithms, computational efficiency.**

I. Introduction

SPARSE coding is a fundamental problem in signal processing that seeks a sparse solution $\boldsymbol{z}$ to the equation

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{z} + \boldsymbol{\varepsilon}, \tag{1}$$

where $\boldsymbol{z} \in \mathbb{R}^D$ is a latent sparse vector, $\boldsymbol{y} \in \mathbb{R}^N$ is an observed measurement vector, $\boldsymbol{\Phi} \in \mathbb{R}^{N \times D}$ is a known dictionary, and $\boldsymbol{\varepsilon} \in \mathbb{R}^N$ is an unknown noise vector.

Sparse Bayesian learning (SBL) is an effective methodology for sparse coding. It has been employed in several popular models, such as sparse Bayesian regression [1], relevance vector machines [2], and Bayesian compressed sensing [3]–[5]. The practical applications of SBL are numerous, encompassing diverse examples such as medical image reconstruction [6], [7], direction of arrival estimation [8], human pose estimation [9], structural health monitoring [10], seismic exploration [11], and visual tracking [12].

SBL offers several advantages compared to other common approaches to sparse coding (e.g. ℓ_0 or ℓ_1 regularization). As a Bayesian method, SBL provides uncertainty quantification and the ability to recover credible intervals over $\boldsymbol{z}$ instead of a single point solution. Moreover, SBL does not need to tune regularization parameters since it can learn them or integrate them out using hyperpriors [4]. As a generative model, SBL can be embedded as a submodule within a larger framework to enforce more complex structure for $\boldsymbol{z}$ (e.g. group sparsity [13], block sparsity [14]). Finally, SBL has favorable optimization properties, such as a sparser global minimum than ℓ_1 methods and fewer local minima than ℓ_0 methods [15].

One often-noted limitation of SBL is the heavy computational cost of its inference algorithm [2], [6]. On the one hand, the fact that SBL requires more computation than non-Bayesian approaches should be unsurprising, since SBL recovers an entire distribution instead of a single point estimate. On the other hand, the most widely-used options for SBL inference scale poorly to high-dimensional problems, which are becoming increasingly common in many domains. This limitation threatens to render SBL obsolete for large-scale settings, as inference cannot be performed in an acceptable timeframe for practical applications.

The principal inference procedure of SBL is the expectation-maximization (EM) algorithm [1], [2]. Each iteration of EM is computationally expensive, requiring $O(D^3)$-time and $O(D^2)$-space to invert a large covariance matrix, where D is the dimension of the sparse codes. There have been several attempts to reduce the costs of EM, using iteratively reweighted least-squares [16], approximate message passing [17], Gaussian belief propagation [18], and variational inference [19]. A popular alternative to EM called the sequential algorithm [20] is also often faster in practice. However, these methods lack either *scalability* at very high dimensions D or *accuracy* in recovering ground-truth sparse codes when compared to EM.

A. Contributions

We introduce a novel method for accelerating SBL's EM algorithm for high-dimensional problems. We call our method covariance-free expectation maximization (CoFEM) because it eliminates the main bottleneck of EM – i.e. the storage and

Manuscript received 28 February 2022; revised 6 June 2022; accepted 7 June 2022. Date of publication 27 June 2022; date of current version 2 August 2022. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yik-chung Wu. This work was supported by National Science Foundation under Cooperative Agreement PHY-2019786. *(Corresponding author: Alexander Lin.)*

Alexander Lin and Demba Ba are with the School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138 USA (e-mail: alin@seas.harvard.edu; demba@seas.harvard.edu).

Andrew H. Song is with the Electrical Engineering and Computer Science Department, Massachusetts Institute of Technology, Cambridge, MA 02138 USA (e-mail: andrew90@mit.edu).

Berkin Bilgic is with the Harvard-MIT Health Sciences and Technology, Massachusetts Institute of Technology, Cambridge, MA 02138 USA, also with the Athinoula A. Martinos Center for Biomedical Imaging, Charlestown, MA 02129 USA, and also with the Department of Radiology, Harvard Medical School, Boston, MA 02115 USA (e-mail: bbilgic@mgh.harvard.edu).

Digital Object Identifier 10.1109/TSP.2022.3186185

inversion of the covariance matrix. We demonstrate that CoFEM has significant advantages in both scalability and accuracy over other SBL approaches (reviewed in Section II). Our contributions are categorized into four types – methodology (Section III), theory (Section IV), experimentation (Section V), and applications (Section VI).[1]

Methodology: We develop CoFEM, which solves multiple linear systems and exploits a little-known diagonal estimation rule to obtain *unbiased* estimates of the posterior moments needed by EM. The multiple systems can be solved in parallel using an iterative solver, such as the conjugate gradient (CG) algorithm, without constructing the covariance matrix. This simple, yet powerful principle reduces EM's per-iteration time complexity from $O(D^3)$ to $O(UK\tau_D)$, where τ_D is the time needed for matrix-vector multiplication and $U, K \ll D$ are hyperparameters for the number of CG steps and the number of linear systems. Furthermore, CoFEM reduces EM's space complexity from $O(D^2)$ to $O(DK)$.

Theory: We prove new theorems that further characterize CoFEM's asymptotic complexities. In applying this theory to matrices $\mathbf{\Phi}$ that satisfy the restricted isometry property (RIP) and commonly occur in compressed sensing applications, we show that, in the limit of a large number of EM iterations, CoFEM's hyperparameters U and K can be kept small even as the dimensionality of the problem D grows very large. We use this theory to justify practical choices for U and K that need not increase with D, leading to signficant computational savings compared to existing SBL algorithms.

Experimentation: We perform simulations comparing CoFEM to several existing SBL algorithms [2], [16], [17], [18], [19], [20], for high-dimensional sparse signal recovery. Across all of our settings, CoFEM is able to maintain the same level of accuracy as EM due to its unbiased estimation of posterior moments. In addition, CoFEM's highly-parallelized and space-saving design enables further acceleration via graphics processing units (GPUs); most other approaches cannot fully realize this benefit due to their heavy memory requirements. In practice, CoFEM with GPU acceleration can reduce hours of computation for covariance-based algorithms to a few seconds.

Applications: We apply CoFEM-equipped SBL in two settings of practical interest: (1) calcium deconvolution and (2) multi-contrast MRI reconstruction. In these high-dimensional settings, CoFEM attains competitive computation time with non-Bayesian approaches, while providing several benefits (e.g. superior performance, uncertainty quantification). These applications require extensions of the SBL model to multi-task learning and to settings with non-negativity constraints, which we demonstrate CoFEM is flexible enough to handle.

II. Sparse Bayesian Learning

A. Generative Model

To solve (1), SBL imposes the following model:

$$\begin{aligned} \boldsymbol{z} &\sim \mathcal{N}(\mathbf{0}, \operatorname{diag}\{\boldsymbol{\alpha}\}^{-1}), \\ \boldsymbol{y} \mid \boldsymbol{z} &\sim \mathcal{N}(\mathbf{\Phi}\boldsymbol{z}, 1/\beta\mathbf{I}), \end{aligned} \tag{2}$$

where $\beta \in \mathbb{R}$ is the inverse of the variance of the noise (commonly called precision) and $\mathbf{I}$ is the identity matrix. Given $\boldsymbol{y}$, SBL performs inference on this model to recover $\boldsymbol{z}$.

The main identifying feature of SBL is the diagonal Gaussian prior with precision parameters $\boldsymbol{\alpha} \in \mathbb{R}^D$ for $\boldsymbol{z}$. The notation $\operatorname{diag}\{\boldsymbol{\alpha}\}$ in (2) maps $\boldsymbol{\alpha}$ to a $D \times D$ matrix with $\boldsymbol{\alpha}$ along its diagonal and zero elsewhere. SBL performs type II maximum likelihood estimation [21] by integrating out $\boldsymbol{z}$ and optimizing $\boldsymbol{\alpha}$. Thus, SBL learns a posterior distribution with uncertainty over $\boldsymbol{z}$. The overall learning objective is:

$$\max_{\boldsymbol{\alpha}} \log p(\boldsymbol{y} \mid \boldsymbol{\alpha}) = \log \int_{\boldsymbol{z}} p(\boldsymbol{y} \mid \boldsymbol{z}) p(\boldsymbol{z} \mid \boldsymbol{\alpha}) d\boldsymbol{z}. \tag{3}$$

As this objective is optimized, many of the elements of $\boldsymbol{\alpha}$ diverge to ∞. For these elements, the independent Gaussian priors converge to point masses at zero, forcing their respective posteriors to follow suit. Thus, upon convergence of $\boldsymbol{\alpha}$ to $\hat{\boldsymbol{\alpha}}$, the recovered posterior distribution $p(\boldsymbol{z} \mid \boldsymbol{y}, \hat{\boldsymbol{\alpha}})$ is often highly sparse. This phenomenon is called *automatic relevance determination* [15] because SBL learns which elements of $\boldsymbol{z}$ are "relevant" (i.e. non-zero) from the data.

In the remainder of this section, we review several inference schemes that have been proposed to optimize (3) and comment on their respective shortcomings.

B. Expectation-Maximization Algorithm

Most SBL inference schemes are built on the *expectation-maximization* (EM) algorithm, a general framework for parameter estimation in the presence of latent variables [2], [22]. EM iteratively alternates between an expectation step (E-Step) and a maximization step (M-Step). Let $\boldsymbol{\alpha}^{(t)}$ be the solution at the start of the t-th iteration. The E-Step computes the expectation $Q(\boldsymbol{\alpha}; \boldsymbol{\alpha}^{(t)})$ of the complete data log-likelihood $\log p(\boldsymbol{z}, \boldsymbol{y} \mid \boldsymbol{\alpha})$ with respect to the latent posterior $p(\boldsymbol{z} \mid \boldsymbol{y}, \boldsymbol{\alpha}^{(t)})$:

$$\begin{aligned} Q(\boldsymbol{\alpha}; \boldsymbol{\alpha}^{(t)}) &= \mathbb{E}_{p(\boldsymbol{z} \mid \boldsymbol{y}, \boldsymbol{\alpha}^{(t)})}[\log p(\boldsymbol{z}, \boldsymbol{y} \mid \boldsymbol{\alpha})] \\ &= \mathbb{E}_{p(\boldsymbol{z} \mid \boldsymbol{y}, \boldsymbol{\alpha}^{(t)})}[\log p(\boldsymbol{z} \mid \boldsymbol{\alpha}) + \log p(\boldsymbol{y} \mid \boldsymbol{z})] \\ &\propto \sum_{j=1}^{D} \log \alpha_j - \alpha_j \cdot \mathbb{E}_{p(\boldsymbol{z} \mid \boldsymbol{y}, \boldsymbol{\alpha}^{(t)})}\left[z_j^2\right] + \text{Const}, \end{aligned} \tag{4}$$

where "Const" absorbs all terms that are constant with respect to $\boldsymbol{\alpha}$. The posterior $p(\boldsymbol{z} \mid \boldsymbol{y}, \boldsymbol{\alpha}^{(t)})$ is a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$ with mean and covariance parameters

$$\boldsymbol{\mu} = \beta \mathbf{\Sigma} \mathbf{\Phi}^\top \boldsymbol{y}, \qquad \mathbf{\Sigma} = (\beta \mathbf{\Phi}^\top \mathbf{\Phi} + \operatorname{diag}\{\boldsymbol{\alpha}^{(t)}\})^{-1}. \tag{5}$$

The second moment of each z_j in (4) can be decomposed into a sum over the squared mean and variance, i.e.

$$\begin{aligned} \mathbb{E}_{p(\boldsymbol{z} \mid \boldsymbol{y}, \boldsymbol{\alpha}^{(t)})}[z_j^2] &= \mathbb{E}_{p(\boldsymbol{z} \mid \boldsymbol{y}, \boldsymbol{\alpha}^{(t)})}[z_j]^2 + \mathbb{V}\text{ar}_{p(\boldsymbol{z} \mid \boldsymbol{y}, \boldsymbol{\alpha}^{(t)})}[z_j] \\ &= \mu_j^2 + \Sigma_{j,j}, \end{aligned} \tag{6}$$

The M-Step updates each $\alpha_j^{(t)}$ by maximizing (4) with respect to α_j. Given $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$, this can be done in closed-form by

[1] Our code can be accessed at https://github.com/al5250/sparse-bayes-learn.

differentiating Q:

$$\frac{\partial Q}{\partial \alpha_j} = \frac{1}{\alpha_j} - \left(\mu_j^2 + \Sigma_{j,j}\right) = 0 \Rightarrow \alpha_j^{(t+1)} = \frac{1}{\mu_j^2 + \Sigma_{j,j}}, \quad (7)$$

EM repeats (5) and (7) for T iterations until convergence, while guaranteeing non-negative change in the log-likelihood objective of (3) at each step.

Despite its simplicity, the EM algorithm is limited by its computational cost. The E-Step of (5) is expensive for large D. Storing $\boldsymbol{\Sigma}$ requires $O(D^2)$-space and computing it through matrix inversion requires $O(D^3)$-time. This makes the standard EM algorithm challenging at high dimensions.

C. Previous Attempts to Accelerate EM

There have been several attempts to accelerate the EM algorithm for SBL inference. One approach is based on *iteratively reweighted least squares* (IRLS) [16], which applies the Woodbury matrix identity to $\boldsymbol{\Sigma}$ in (5), yielding

$$\boldsymbol{\Sigma} = \mathbf{C} - \mathbf{C}\boldsymbol{\Phi}^\top \left(\mathbf{I} + \boldsymbol{\Phi}\mathbf{C}\boldsymbol{\Phi}^\top\right)^{-1} \boldsymbol{\Phi}\mathbf{C}, \quad (8)$$

where $\mathbf{C} = \text{diag}\{\boldsymbol{\alpha}^{(t)}\}^{-1}$. By inverting an $N \times N$ matrix (as opposed to a $D \times D$ matrix), IRLS reduces the per-iteration time complexity of EM to $O(N^3 + N^2 D)$. For problems in which $N \ll D$, IRLS is more efficient than EM. However, if $N = O(D)$, the time complexity is still $O(D^3)$. Furthermore, like EM, IRLS requires storage of $\boldsymbol{\Sigma}$, which remains an expensive $O(D^2)$-space cost.

Another line of work is based on variants of *approximate message passing* (AMP) [17] and *Gaussian belief propagation* (GBP) [18]. Within each E-Step, these methods run algorithms comprised of T_{amp} and T_{gbp} inner steps to estimate the means and variances of $\boldsymbol{z}$ in (6), thereby circumventing matrix inversion. Though they can be faster than EM in practice, AMP and GBP may fail to converge when estimating variances in certain cases [23], for example if the dictionary $\boldsymbol{\Phi}$ does not satisfy zero-mean and sub-Gaussian criteria [24].

An alternative strategy is to employ *variational inference* (VI), which approximates the true posterior $p(\boldsymbol{z} \mid \boldsymbol{y}, \boldsymbol{\alpha})$ with a simpler surrogate $q(\boldsymbol{z})$ [25]–[27]. This allows for SBL inference that is inverse-free [19], [28]. However, VI approaches optimize a lower bound on (3) instead of the true objective. Thus, they may converge to a sub-optimal solution for $\boldsymbol{\alpha}$.

AMP, GBP, and VI are all limited by the fact that their approximations to the means and variances of $\boldsymbol{z}$ can be *biased* for general dictionaries $\boldsymbol{\Phi}$ [24], [28]. In Section III, we present a new method that ensures an *unbiased* estimation of these moments, regardless of the structure of $\boldsymbol{\Phi}$.

D. Sequential Algorithm

The sequential (Seq) algorithm, is a notable alternative to EM that reduces computation time and space in practice [3], [20]. Seq maintains a set $\mathcal{S} \subseteq \{1, 2, \ldots, D\}$ of "active" indices such that α_j is finite for each $j \in \mathcal{S}$ and $\alpha_j = \infty$ for all $j \neq \mathcal{S}$. Initially, $\mathcal{S} = \emptyset$. Indices are sequentially added to or deleted from $\mathcal{S}$ if making such a change can increase the log-likelihood objective (3).

At any given point, Seq only needs to store parts of the mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ corresponding to $\mathcal{S}$; all other components are assumed to be zero. Thus, for truly sparse vectors $\boldsymbol{z}$ with d non-zero components such that $d \ll D$, Seq is more efficient than EM. Yet unlike EM, the number of iterations needed for Seq depends on d, since at least d steps must be taken to fully recover $\boldsymbol{z}$. The overall time complexity is $O(d^2 D)$ and the space complexity is $O(d^2 + D)$ [4].

However, Seq has several limitations. It is often the case that d is a fraction or percentage of D. If d grows linearly with D, the asymptotic time cost is $O(D^3)$, similar to EM. This may explain why Seq can still be up to hundreds of times slower than non-Bayesian methods for large D [6]. Also, the algorithm's sequential nature hinders its potential for speedup on parallel machines. Lastly, for large D, it remains costly to store parts of the quadratically-sized covariance matrix $\boldsymbol{\Sigma}$.

III. Covariance-Free Expectation-Maximization

We introduce *covariance-free expectation-maximization* (CoFEM), a new SBL inference scheme for accelerating EM that removes the need to compute (let alone invert) the covariance matrix $\boldsymbol{\Sigma}$. Our method is built on several advances from numerical linear algebra [29], [30].

Our key observation is that not all elements of $\boldsymbol{\Sigma}$ are needed for the M-Step in (7). Indeed, we only need $\boldsymbol{\mu}$ (which depends on $\boldsymbol{\Sigma}$ via (5)) and the the diagonal elements of $\boldsymbol{\Sigma}$ to update $\boldsymbol{\alpha}$. Thus, we propose a simplified E-Step that can estimate the two desired quantities $\{\mu_j, \Sigma_{j,j}\}$ for all j from *solutions to linear systems*, thereby avoiding the need for matrix inversion. We can re-express (5) for $\boldsymbol{\mu}$ as

$$\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} = \beta \boldsymbol{\Phi}^\top \boldsymbol{y}, \quad (9)$$

where $\boldsymbol{\Sigma}^{-1} = \beta \boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \text{diag}\{\boldsymbol{\alpha}^{(t)}\}$. Thus, $\boldsymbol{\mu}$ is the solution $\boldsymbol{x}$ to the linear system $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$ for $\mathbf{A} := \boldsymbol{\Sigma}^{-1}$ and $\boldsymbol{b} := \beta \boldsymbol{\Phi}^\top \boldsymbol{y}$.

A. Diagonal Estimation

We leverage a result from [29] to obtain the diagonal of $\boldsymbol{\Sigma}$.

Proposition 1 (Diagonal Estimation Rule): Let $\mathbf{M} \in \mathbb{R}^{D \times D}$ be a square matrix. Let $\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_K \in \mathbb{R}^D$ be random probe vectors, where each $\boldsymbol{p}_k$ comprises independent and identically distributed elements such that $\mathbb{E}[p_{k,j}] = 0$ for all $j = 1, \ldots, D$. For each $\boldsymbol{p}_k$, let $\boldsymbol{x}_k = \mathbf{M}\boldsymbol{p}_k$. Consider the estimator $\boldsymbol{s} \in \mathbb{R}^D$, where, for each $j = 1, \ldots, D$,

$$s_j = \frac{\sum_{k=1}^K p_{k,j} \cdot x_{k,j}}{\sum_{k=1}^K p_{k,j}^2}. \quad (10)$$

Then, each s_j is an unbiased estimator of $\mathrm{M}_{j,j}$.

Proof: Consider s_j, the j-th element of $\boldsymbol{s}$. We have

$$s_j = \frac{\sum_{k=1}^K \left(p_{k,j} \cdot \left(\sum_{j'=1}^D \mathrm{M}_{j,j'} \cdot p_{k,j'}\right)\right)}{\sum_{k=1}^K p_{k,j}^2}$$

$$= \mathrm{M}_{j,j} + \sum_{j' \neq j} \mathrm{M}_{j,j'} \cdot \frac{\sum_{k=1}^K p_{k,j} \cdot p_{k,j'}}{\sum_{k=1}^K p_{k,j}^2}. \quad (11)$$

Thus, $\mathbb{E}[s_j]$ is equal to the following:

$$\mathrm{M}_{j,j} + \sum_{j' \neq j} \mathrm{M}_{j,j'} \cdot \left(\sum_{k=1}^{K} \underbrace{\mathbb{E}[p_{k,j'}]}_{=0} \cdot \mathbb{E}\left[\frac{p_{k,j}}{\sum_{k=1}^{K} p_{k,j}^2} \right] \right), \quad (12)$$

where we have applied the fact that the j and j' components of $\boldsymbol{p}_k$ are independent to arrive at a product of expectations. Since $\mathbb{E}[p_{k,j'}] = 0$ for all k and j', it follows that $\mathbb{E}[s_j] = \mathrm{M}_{j,j}$. ■

We apply this rule to $\boldsymbol{\Sigma}$ to estimate its diagonal elements. Following [29], we use the *Rademacher distribution* for drawing the probe vector $\boldsymbol{p}_k$, where each $p_{k,j}$ is either -1 or $+1$ with equal probability. This simplifies (10) to

$$s_j = \frac{1}{K} \sum_{k=1}^{K} p_{k,j} \cdot x_{k,j}. \quad (13)$$

The diagonal estimation rule turns an *explicit* property of a matrix (i.e. the diagonal elements) into an *implicit* quantity that can be estimated without physically forming the matrix. To exploit this rule, we only need a method for applying $\boldsymbol{\Sigma}$ to each vector $\boldsymbol{p}_k$ to obtain $\boldsymbol{x}_k$; this can be done by solving a linear system $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$, where $\mathbf{A} := \boldsymbol{\Sigma}^{-1}$ and $\boldsymbol{b} := \boldsymbol{p}_k$.

In summary, the quantities $\{\mu_j, \Sigma_{j,j}\}$ needed for the simplified E-Step update can be obtained by solving $K+1$ separate linear systems. These systems can be solved in parallel by considering the matrix equation $\mathbf{A}\mathbf{X} = \mathbf{B}$ with inputs $\mathbf{A} \in \mathbb{R}^{D \times D}$ and $\mathbf{B} \in \mathbb{R}^{D \times (K+1)}$ defined as follows:

$$\mathbf{A} := \beta \boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \mathrm{diag}\{\boldsymbol{\alpha}^{(t)}\},$$
$$\mathbf{B} := \left[\boldsymbol{p}_1 \mid \boldsymbol{p}_2 \mid \ldots \mid \boldsymbol{p}_K \mid \beta \boldsymbol{\Phi}^\top \boldsymbol{y} \right]. \quad (14)$$

If we enumerate the columns of the solution matrix $\mathbf{X} \in \mathbb{R}^{D \times (K+1)}$ as $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_K, \boldsymbol{\mu}$, then our desired quantities for the simplified E-Step are $\boldsymbol{\mu}$ and $\boldsymbol{s}$, as calculated by (13). We can then perform the M-Step update in (7) as

$$\alpha_j^{(t+1)} = \frac{1}{\mu_j^2 + s_j}, \quad (15)$$

completely avoiding the need to compute or invert the covariance matrix $\boldsymbol{\Sigma}$. Algorithm 1 gives the full CoFEM algorithm.

The diagonal estimator in (13) is unbiased for any $K \geq 1$, yet its variance is proportional to $1/K$, as we will show in Section IV-A. We will provide theoretical justification that small K suffices for CoFEM in practice, and that this K can be constant with respect to the dimensionality D for a large class of compressed sensing matrices $\boldsymbol{\Phi}$.

Algorithm 1: CovarianceFreeEM($\boldsymbol{y}$, $\boldsymbol{\Phi}$, β, T, K).

1: Initialize $\alpha_j^{(1)} \leftarrow 1$ for $j = 1, \ldots, D$.
2: **for** $t = 1, 2, \ldots, T$ **do**
3: *// Simplified E-Step*
4: Define $\mathbf{A} \leftarrow \beta \boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \mathrm{diag}\{\boldsymbol{\alpha}^{(t)}\}$.
5: Draw $\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_K \sim$ Rademacher distribution.
6: Define $\mathbf{B} \leftarrow [\boldsymbol{p}_1 \mid \boldsymbol{p}_2 \mid \ldots \mid \boldsymbol{p}_K \mid \beta \boldsymbol{\Phi}^\top \boldsymbol{y}]$.
7: $[\boldsymbol{x}_1 \mid \boldsymbol{x}_2 \mid \ldots \mid \boldsymbol{x}_K \mid \boldsymbol{\mu}] \leftarrow$ LinearSolver($\mathbf{A}, \mathbf{B}$).
8: Compute $s_j \leftarrow 1/K \sum_{k=1}^{K} p_{k,j} \cdot x_{k,j}$ for $j = 1, \ldots, D$.
9: *// M-Step*
10: **if** $t < T$ **then**
11: Update $\alpha_j^{(t+1)} \leftarrow 1/(\mu_j^2 + s_j)$ for $j = 1, \ldots, D$.
12: **end if**
13: **end for**
14: **return** $\boldsymbol{\alpha}^{(T)}, \boldsymbol{\mu}, \boldsymbol{s}$

B. Parallel Conjugate Gradient

Among potential options for the linear solver in Algorithm 1, we choose the conjugate gradient (CG) algorithm due to its efficiency and flexibility [30], [31]. CG is an iterative approach with a series of matrix-vector multiplication steps to solve a linear system $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$. At each step, CG does not need a physical manifestation of $\mathbf{A}$; it only requires a way to apply $\mathbf{A}$ to an arbitrary vector $\boldsymbol{v}$ to yield $\mathbf{A}\boldsymbol{v}$. For SBL, $\mathbf{A} := \beta \boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \mathrm{diag}\{\boldsymbol{\alpha}^{(t)}\}$, which implies the complexity of CG is dominated by the time τ_D it takes to apply $\boldsymbol{\Phi}$ and its transpose to $\boldsymbol{v}$. In many applications, $\boldsymbol{\Phi}$ is a structured and matrix-free operation. Examples include convolution, discrete cosine transform, Fourier transform, and wavelet transform – all of which require at most $\tau_D = O(D \log D)$-time [32].

CG can also easily generalize to solving *multiple* linear systems $\mathbf{A}\mathbf{X} = \mathbf{B}$ by simply replacing the matrix-vector multiplications with matrix-matrix multiplications. For faster computing, these operations can be parallelized on GPUs. In addition, CG is space-efficient and only needs $O(D)$-space to solve the linear system; this is the minimum requirement for any solver given that the output $\boldsymbol{x} \in \mathbb{R}^D$.

C. Preconditioning

The time complexity of CG depends on the number of CG steps U. It is guaranteed that $U \leq D$ [30], yet D can be very large for high-dimensional problems. In general, if $\mathbf{A}$ has a small condition number $\kappa(\mathbf{A}) := \lambda_{\max}(\mathbf{A})/\lambda_{\min}(\mathbf{A})$ where $\lambda_{\max}(\mathbf{A})$ and $\lambda_{\min}(\mathbf{A})$ are the largest and smallest eigenvalues of $\mathbf{A}$, then $U \ll D$ steps are needed to find an $\hat{\boldsymbol{x}}$ such that $\|\mathbf{A}\hat{\boldsymbol{x}} - \boldsymbol{b}\|_2 / \|\boldsymbol{b}\|_2 \leq \epsilon_{\max}$ for small $\epsilon_{\max}$. However, optimizing the SBL objective function pushes many diagonal elements of $\mathbf{A}$ to ∞, resulting in large $\kappa(\mathbf{A})$ and necessitating large U.

To resolve this issue, we incorporate a *preconditioner* matrix $\mathbf{M} \in \mathbb{R}^{D \times D}$ in CG. Instead of directly solving $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$, we solve an equivalent system $\mathbf{A}'\boldsymbol{x}' = \boldsymbol{b}'$, where $\mathbf{A}' := \mathbf{M}^{-1/2}\mathbf{A}\mathbf{M}^{-1/2}$, $\boldsymbol{b}' := \mathbf{M}^{-1/2}\boldsymbol{b}$ and $\boldsymbol{x}' := \mathbf{M}^{1/2}\boldsymbol{x}$. We consider two criteria for choosing $\mathbf{M}$ in CoFEM. First, we want $\kappa(\mathbf{A}') \ll \kappa(\mathbf{A})$ to reduce the number of steps U that are necessary to achieve small $\epsilon_{\max}$. Next, we want to maintain the scalability of CoFEM (e.g. absence of matrix inversion, $O(D)$-space complexity) by avoiding a dense matrix for $\mathbf{M}$.

We thus propose to use a *diagonal preconditioner* $\mathbf{M} = \mathrm{diag}\{\beta\boldsymbol{\theta} + \boldsymbol{\alpha}^{(t)}\}$, where $\boldsymbol{\theta} \in \mathbb{R}^D$ is a set of customizable positive values. This $\mathbf{M}$ is easy to invert, only requires $O(D)$-space, and can be quickly applied to vectors. By varying $\boldsymbol{\theta}$, we can arrive at different choices for $\mathbf{M}$. For example, setting $\theta_j = \sum_{i=1}^{N} \Phi_{i,j}^2$ leads to the popular *Jacobi preconditioner* satisfying

Algorithm 2: ParallelConjGradient($\mathbf{A}, \mathbf{B}, \mathbf{M}, \mathbf{U}, \epsilon_{\max}$).

1: Initialize $\mathbf{X}$ as a $D \times Q$ matrix of all zeros.
2: Initialize $\mathbf{R} \leftarrow \mathbf{B}$ and $\mathbf{P} \leftarrow \mathbf{B}$ and $\mathbf{W} \leftarrow \mathbf{M^{-1}B}$.
3: Compute $\rho_q \leftarrow \sum_{j=1}^{D} \mathrm{R}_{j,q} \cdot \mathrm{W}_{j,q}$ for $q = 1, \ldots, Q$.
4: **for** $u = 1, 2, \ldots, U$ **do**
5: Compute $\mathbf{\Psi} \leftarrow \mathbf{AP}$. *// Apply matrix.*
6: Compute $\pi_q \leftarrow \sum_{j=1}^{D} \mathrm{P}_{d,q} \cdot \Psi_{d,q}$ for $q = 1, \ldots, Q$.
7: Compute $\gamma_q \leftarrow \rho_q / \pi_q$ for $q = 1, \ldots, Q$.
8: Update $\mathbf{X} \leftarrow \mathbf{X} + \mathbf{P\Gamma}$, where $\mathbf{\Gamma} = \mathrm{diag}\{\boldsymbol{\gamma}\}$.
9: Update $\mathbf{R} \leftarrow \mathbf{R} - \mathbf{\Psi\Gamma}$, where $\mathbf{\Gamma} = \mathrm{diag}\{\boldsymbol{\gamma}\}$.
10: Let $\delta \leftarrow ||\mathbf{R}||_F / ||\mathbf{B}||_F$, where F is Frobenius norm.
11: **if** $\delta \leq \epsilon_{\max}$ **then**
12: **returnX**
13: **end if**
14: Compute $\mathbf{W} \leftarrow \mathbf{M^{-1}R}$. *// Apply preconditioner.*
15: Let $\rho_q^{\text{old}} \leftarrow \rho_q$ for $q = 1, \ldots, Q$.
16: Compute $\rho_q \leftarrow \sum_{j=1}^{D} \mathrm{R}_{j,q} \cdot \mathrm{W}_{j,q}$ for $q = 1, \ldots, Q$.
17: Compute $\eta_q \leftarrow \rho_q / \rho_q^{\text{old}}$ for $q = 1, \ldots, Q$.
18: Update $\mathbf{P} \leftarrow \mathbf{R} + \mathbf{PH}$, where $\mathbf{H} = \mathrm{diag}\{\boldsymbol{\eta}\}$.
19: **end for**
20: **return X**

TABLE I
COMPUTATIONAL COMPLEXITIES OF SBL INFERENCE SCHEMES

Method	**Time** (per iter)	Space	EM-Based
EM [2]	$O(D^3)$	$O(D^2)$	✓
IRLS [16]	$O(DN^2 + N^3)$	$O(D^2)$	✓
AMP [17]	$O(DNT_{\text{amp}})$	$O(DN)$	✓
GBP [18]	$O(DNT_{\text{gbp}})$	$O(DN)$	✓
VI [19]	$O(\tau_D)$	$O(D)$	✓
Seq [20]	$O(Dd)$	$O(D + d^2)$	✗
CoFEM (ours)	$O(\tau_D UK)$	$O(DK)$	✓

$\mathrm{M}_{j,j} = \mathrm{A}_{j,j}$ [31]. In Section IV-B, we provide novel theoretical analysis for our diagonal preconditioner within the context of SBL. We illustrate that for a large class of compressed sensing dictionaries $\mathbf{\Phi}$, setting $\theta_j = 1$ for all j is a favorable choice that leads to small $\kappa(\mathbf{A}')$ and enables U to be constant with respect to the dimensionality D.

Algorithm 2 summarizes the parallel CG algorithm for inputs $\mathbf{A} \in \mathbb{R}^{D \times D}$, $\mathbf{M} \in \mathbb{R}^{D \times D}$, and $\mathbf{B} \in \mathbb{R}^{D \times Q}$, where Q is the number of parallel systems. For CoFEM, we have $Q = K + 1$. The computation is dominated by line 5, in which $\mathbf{A}$ is applied to vectors stored as columns of a matrix.

D. Complexity Comparisons

Each of the T iterations of CoFEM requires at most U steps of CG – in which we apply $\mathbf{\Phi}$ (and $\mathbf{\Phi}^\top$) in τ_D-time to K vectors – giving us an overall time complexity of $O(T\tau_D UK)$. CoFEM's space complexity is dominated by CG, which requires $O(D)$-space for each of the $(K + 1)$ systems. Table I shows the per-iteration complexities of CoFEM and other SBL inference schemes. The notion of a single iteration differs for the sequential algorithm, compared to the other methods, as it does not use EM. While many other methods improve upon EM, they introduce dependencies on N or d, which typically grow with D. For example, if the size of the signal $\boldsymbol{z}$ is doubled, we may also expect the number of measurements N to be doubled (to achieve same reconstruction error), as well as the number d of non-zero values in $\boldsymbol{z}$. Thus, increasing D compounds the increase in complexities of these algorithms. In contrast, CoFEM's dependencies on U and K can be held constant as D increases, which we demonstrate in Section V.

E. CoFEM for SBL Extensions

To further highlight the flexibility of CoFEM, we show how it can handle two common extensions of the SBL model.

1) Multi-Task Learning: In multi-task learning, there are L different sparse vector recovery problems that one wishes to solve at once. These problems may have different observation-dictionary pairs $(\boldsymbol{y}_1, \mathbf{\Phi}_1), (\boldsymbol{y}_2, \mathbf{\Phi}_2), \ldots, (\boldsymbol{y}_L, \mathbf{\Phi}_L)$, yet the tasks are related in the sense that their corresponding vectors $\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_L$ have similar non-zero supports. Some examples include multiple measurements vector (MMV) problems [33], multi-task compressed sensing [4], and sparse Bayesian learning with complex numbers [34].

A simple way to enforce joint sparsity among all tasks in SBL is to have them share a common $\boldsymbol{\alpha}$ vector:

$$\begin{aligned} \boldsymbol{z}_\ell &\sim \mathcal{N}(\mathbf{0}, \mathrm{diag}\{\boldsymbol{\alpha}\}^{-1}), && \ell = 1, 2, \ldots, L, \\ \boldsymbol{y}_\ell &\sim \mathcal{N}(\mathbf{\Phi}_\ell \boldsymbol{z}_\ell, 1/\beta \mathbf{I}), && \ell = 1, 2, \ldots, L. \end{aligned} \tag{16}$$

Learning takes place through the task-separable objective:

$$\max_{\boldsymbol{\alpha}} \log p(\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_L \mid \boldsymbol{\alpha}) = \sum_{\ell=1}^{L} \log p(\boldsymbol{y}_\ell \mid \boldsymbol{\alpha}). \tag{17}$$

To optimize (17), EM runs a E-Step for each task ℓ,

$$\boldsymbol{\mu}_\ell = \beta \mathbf{\Sigma}_\ell \mathbf{\Phi}_\ell^\top \boldsymbol{y}_\ell, \quad \mathbf{\Sigma}_\ell = (\beta \mathbf{\Phi}_\ell^\top \mathbf{\Phi}_\ell + \mathrm{diag}\{\boldsymbol{\alpha}^{(t)}\})^{-1}, \tag{18}$$

followed by a M-Step to combine these moments,

$$\alpha_j^{(t+1)} = \frac{L}{\sum_{\ell=1}^{L} \mu_{\ell,j}^2 + \Sigma_{\ell,j,j}}. \tag{19}$$

To accelerate EM, CoFEM can simply replace the E-Step for each task ℓ with a covariance-free version, as described in Section III-A. We can further parallelize these L E-Steps by solving their $L \cdot (K + 1)$ systems all at once through Alg. 2.

2) Non-Negativity Constraints: In some applications, we may want to enforce non-negativity on $\boldsymbol{z}$. In these cases, we can use an independent *rectified Gaussian* prior $\mathcal{N}^R(0, 1/\alpha_j)$ for each component z_j of $\boldsymbol{z}$ [35], which places zero probability mass on the negative values. Specifically, we have

$$p(z_j \mid \alpha_j) = \begin{cases} \sqrt{2\alpha_j/\pi} \exp(-\alpha_j z_j^2/2), & z_j > 0, \\ 1/2, & z_j = 0, \\ 0, & z_j < 0. \end{cases} \tag{20}$$

Due to conjugacy between the rectified Gaussian and Gaussian distributions, the posterior $p(\boldsymbol{z} \mid \boldsymbol{y}, \hat{\boldsymbol{\alpha}})$ is also a rectified Gaussian. However, this posterior's density function is not analytically tractable, so we can follow [35] and approximate it

with a diagonal rectified Gaussian whose second moment is

$$\mathbb{E}_{p(\boldsymbol{z}\,|\,\boldsymbol{y},\hat{\boldsymbol{\alpha}})}[z_j^2] = \mu_j^2 + \Sigma_{j,j} + \mu_j \cdot \sqrt{\frac{\Sigma_{j,j}}{\pi}} \cdot \frac{\exp(-\xi_j^2)}{\operatorname{erfc}(-\xi_j)}, \quad (21)$$

where $\xi_j = \mu_j / \sqrt{2\Sigma_{j,j}}$ and $\operatorname{erfc}(x) = 2/\sqrt{\pi} \int_x^\infty \exp(-t^2) dt$ is the complimentary error function. To compute (21), CoFEM can replace $\Sigma_{j,j}$ with s_j from (13). The M-Step then updates α_j as the reciprocal of (21).

IV. Theoretical Analysis of CoFEM

The two main hyperparameters of CoFEM are the number of probe vectors K and the number of conjugate gradient steps U. These values determine the per-iteration time complexity of CoFEM as $O(UK\tau_D)$ and its space complexity as $O(DK)$. Thus, it is important to control U and K. In this section, we present new theoretical results illustrating that, for a large class of dictionaries $\boldsymbol{\Phi}$, these hyperparameters can be kept small even as the dimensionality of the problem D grows very large.

The ultimate goal of SBL is to solve a Bayesian variant of the sparse coding problem in which we (a) identify which $z_j = 0$ and (b) provide uncertainty quantification for the non-zero z_j. To achieve true sparsity for a particular z_j, it is necessary for its prior parameter $\alpha_j \to \infty$, which is only attained as the number of iterations $t \to \infty$. Thus, our study of U and K is based on the following SBL Convergence property:

Definition 1 (SBL Convergence): In Alg. 1, consider the sequence of iterates $\boldsymbol{\alpha}^{(t)}$ for $t = 1, 2, \ldots,$ and let $\hat{\boldsymbol{\alpha}} := \lim_{t\to\infty} \boldsymbol{\alpha}^{(t)}$. Then, $(\mathcal{S}, \mathcal{U}, \hat{\boldsymbol{\alpha}})$-convergence is satisfied for SBL if the indices $\mathbb{N}_D := \{1, 2, \ldots, D\}$ can be partitioned into an "active" set $\mathcal{S} \subseteq \mathbb{N}_D$ and an "inactive" set $\mathcal{U} := \mathbb{N}_D \setminus \mathcal{S}$, where $\hat{\alpha}_j > 0$ is finite if $j \in \mathcal{S}$ and $\hat{\alpha}_j = \infty$ if $j \in \mathcal{U}$.

Of course, it is impossible to run $t \to \infty$ iterations in practice and to the best of the authors' knowledge, there does not exist a formal proof characterizing EM's behavior after finitely many iterations in the context of SBL. However, many works within the SBL literature [2]–[5] have illustrated (and even relied on) a well-accepted phenomenon in which the inactive α_j grow very large and can be treated as "reaching infinity" after T finite iterations. This justifies the practical applicability of Definition 1 and our ensuing theoretical analysis.

Notation: For any positive integer P, let $\mathbb{N}_P := \{1, 2, \ldots, P\}$. For any vector $\boldsymbol{v} \in \mathbb{R}^P$ and set $\mathcal{B} \subseteq \mathbb{N}_P$, define the sub-vector $\boldsymbol{v}_\mathcal{B} := [v_b \,|\, b \in \mathcal{B}] \in \mathbb{R}^{|\mathcal{B}|}$. Similarly, for any matrix $\mathbf{M} = [\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(P)}] \in \mathbb{R}^{Q\times P}$, define the sub-matrix $\mathbf{M}_\mathcal{B} := [\boldsymbol{v}^{(b)} \,|\, b \in \mathcal{B}] \in \mathbb{R}^{Q\times|\mathcal{B}|}$. For any two sets $\mathcal{A} \subseteq \mathbb{N}_Q$ and $\mathcal{B} \subseteq \mathbb{N}_P$, define the sub-matrix block $\mathbf{M}_{\mathcal{A},\mathcal{B}} := [\boldsymbol{v}_\mathcal{A}^{(b)} \,|\, b \in \mathcal{B}] \in \mathbb{R}^{|\mathcal{A}|\times|\mathcal{B}|}$. Let $\|\boldsymbol{v}\|_2$ denote the Euclidean norm of vector $\boldsymbol{v}$ and $\|\boldsymbol{v}\|_\mathbf{M} := \sqrt{\boldsymbol{v}^\top \mathbf{M} \boldsymbol{v}}$ denote the $\mathbf{M}$-weighted norm of $\boldsymbol{v}$. Let $\|\mathbf{M}\|_2$ denote the spectral norm (i.e. largest singular value) of matrix $\mathbf{M}$. Let $\sigma_{\min}(\mathbf{M})$, $\lambda_{\max}(\mathbf{M})$ and $\lambda_{\min}(\mathbf{M})$ denote the smallest singular value, largest eigenvalue, and smallest eigenvalue of $\mathbf{M}$, respectively. For a matrix $\mathbf{M}$, let $\kappa(\mathbf{M}) := \|\mathbf{M}\|_2 / \sigma_{\min}(\mathbf{M})$ be the condition number of $\mathbf{M}$.

A. A Theory for the Number of Probe Vectors K

First, we analyze the dependency of the diagonal estimator $\boldsymbol{s}$ on the number of probe vectors K. We aim to characterize the standard deviation (i.e. the square root of the variance) of each s_j, which leads to this lemma:

Lemma 1: Let $\mathbf{M} \in \mathbb{R}^{D\times D}$. Consider the estimator $\boldsymbol{s}$ defined in Prop. 1, where $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_K$ are independent Rademacher variables. Then, the standard deviation ν_j of s_j is

$$\nu_j := \sqrt{\mathbb{E}[(s_j - \mathbb{E}[s_j])^2]} = \sqrt{\frac{1}{K} \sum\nolimits_{j'\neq j} \mathrm{M}_{j,j'}^2}. \quad (22)$$

Proof: Within (22), we substitute the expression for s_j from (11) and the fact that $\mathbb{E}[s_j] = \mathrm{M}_{j,j}$ to yield

$$\begin{aligned}\nu_j &= \sqrt{\mathbb{E}\left[\left(\sum\nolimits_{j'\neq j} \mathrm{M}_{j,j'} \cdot \frac{\sum_{k=1}^K p_{k,j} \cdot p_{k,j'}}{\sum_{k=1}^K p_{k,j}^2}\right)^2\right]} \\ &= \sqrt{\mathbb{E}\left[\sum\nolimits_{j'\neq j} \sum\nolimits_{j''\neq j} \mathrm{M}_{j,j'} \cdot \mathrm{M}_{j,j''} \cdot \frac{g_{j,j'}}{g_{j,j}} \cdot \frac{g_{j,j''}}{g_{j,j}}\right]}, \quad (23)\end{aligned}$$

where we define $g_{j,\ell} := \sum_{k=1}^K p_{k,j} \cdot p_{k,\ell}$ for all $(j, \ell) \in \mathbb{N}_D \times \mathbb{N}_D$. In the denominator, we have $g_{j,j} = K$ for all j since $p^2 = 1$ for a Rademacher variable. In the numerator, if $j' = j''$, we have $\mathbb{E}[g_{j,j'} \cdot g_{j,j''}] = \mathbb{E}[g_{j,j'}^2] = K$. Otherwise, if $j \neq j''$, $\mathbb{E}[g_{j,j'} \cdot g_{j,j''}] = 0$. Thus, (23) simplifies to (22). ■

Lemma 1 tells us that the standard deviation of our estimator decreases with K and increases with the norm of the off-diagonal entries. Analyzing Lemma 1 within the context of SBL leads to our first main theoretical result, as stated below.

Theorem 1: Let $\boldsymbol{\Sigma}^{(t)} := (\beta \boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \operatorname{diag}\{\boldsymbol{\alpha}^{(t)}\})^{-1}$ be the SBL covariance matrix at the t-th iteration of Alg. 1. Let $\boldsymbol{s}^{(t)}$ be the Rademacher diagonal estimator for $\boldsymbol{\Sigma}^{(t)}$ defined in (13) with K probe vectors. Let $\nu_j^{(t)}$ denote the standard deviation of $s_j^{(t)}$. Assume that $(\mathcal{S}, \mathcal{U}, \hat{\boldsymbol{\alpha}})$-convergence is satisfied. Then, for any inactive index $j \in \mathcal{U}$, we have

$$\lim_{t\to\infty} \nu_j^{(t)} = 0, \quad (24)$$

and for any active index $j \in \mathcal{S}$, we have

$$\lim_{t\to\infty} \nu_j^{(t)} \leq \frac{1}{\sqrt{K}} \cdot \frac{\inf_{\boldsymbol{\Theta}\in\mathcal{O}} \|\boldsymbol{\Theta}^{-1} \boldsymbol{\Phi}_\mathcal{S}^\top \boldsymbol{\Phi}_\mathcal{S} - \mathbf{I}\|_2}{\beta \cdot \sigma_{\min}^2(\boldsymbol{\Phi}_\mathcal{S})}, \quad (25)$$

where $\mathcal{O}$ is the set of $|\mathcal{S}| \times |\mathcal{S}|$ diagonal matrix with positive diagonal elements and $\mathbf{I}$ is the identity matrix.

Theorem 1 offers several insights in the limit of EM iterations: (1) the estimator becomes deterministic with zero standard deviation for the inactive indices, (2) K only affects the estimator's standard deviation for the active indices, and (3) if $\boldsymbol{\Phi}_\mathcal{S}^\top \boldsymbol{\Phi}_\mathcal{S}$ is close to any diagonal matrix $\boldsymbol{\Theta}$ (i.e. the columns of $\boldsymbol{\Phi}_\mathcal{S}$ are close to orthogonal), then the standard deviation for the active indices converge to a small quantity. The proof of Theorem 1 is given in Appendix A.

B. A Theory for the Number of Conjugate Gradient Steps U

Next, we analyze the number of CG steps U needed for convergence. We build on the following well-known result [31]:

Lemma 2 (CG Convergence): Consider the CG algorithm for solving $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$, where $\mathbf{A} \in \mathbb{R}^{D\times D}$ is a positive definite matrix and $\boldsymbol{b} \in \mathbb{R}^D$. Let $\boldsymbol{x}_0 \in \mathbb{R}^D$ be the initial solution. Let $\boldsymbol{x}_u$ denote the algorithm's solution and $\boldsymbol{r}_u := \boldsymbol{b} - \mathbf{A}\boldsymbol{x}_u$ denote the algorithm's residual at the u-th step of CG. Then,

$$\|\boldsymbol{r}_u\|_{\mathbf{A}^{-1}} \leq 2\left(\frac{\sqrt{\kappa(\mathbf{A})}-1}{\sqrt{\kappa(\mathbf{A})}+1}\right)^u \|\boldsymbol{r}_0\|_{\mathbf{A}^{-1}}, \tag{26}$$

where $\|\boldsymbol{r}\|_{\mathbf{A}^{-1}} := \sqrt{\boldsymbol{r}^\top \mathbf{A}^{-1}\boldsymbol{r}}$ for any vector $\boldsymbol{r} \in \mathbb{R}^D$.

Corollary 1: Let CG with matrix $\mathbf{A}$ and any $\boldsymbol{b} \in \mathbb{R}^D$ start with the initialization $\boldsymbol{x}_0 = \mathbf{0}$. Let $\epsilon := \|\boldsymbol{r}_U\|_{\mathbf{A}^{-1}}/\|\boldsymbol{b}\|_{\mathbf{A}^{-1}}$ denote the relative residual error of CG after U steps.[2] Then,

$$\epsilon \leq 2\exp(-U/\sqrt{\kappa(\mathbf{A})}). \tag{27}$$

Proof: Since $\boldsymbol{r}_0 = \boldsymbol{b}$, we can apply (26) to obtain

$$\epsilon \leq 2\left(\frac{\sqrt{\kappa(\mathbf{A})}-1}{\sqrt{\kappa(\mathbf{A})}+1}\right)^U \leq 2\left(1-\frac{1}{\sqrt{\kappa(\mathbf{A})}}\right)^U. \tag{28}$$

The inequality $1 - 1/x \leq \exp(-1/x)$ holds for any $x \in \mathbb{R}$. Taking $x = \sqrt{\kappa(\mathbf{A})}$ in (28) gives the result. ■

Corollary 1 indicates that the relative residual error of CG decreases exponentially with U, yet the precise exponent depends on $\kappa(\mathbf{A})$. In the t-th iteration of CoFEM, we wish to solve linear systems with $\mathbf{A}^{(t)} := \beta\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \text{diag}\{\boldsymbol{\alpha}^{(t)}\}$. However, since $\boldsymbol{\alpha}_j^{(t)} \to \infty$ for $j \in \mathcal{U}$, this leads to $\kappa(\mathbf{A}^{(t)}) \to \infty$ and a vacuous bound on ϵ for any finite number of steps U. Therefore, CoFEM introduces a preconditioner $\mathbf{M}$ to construct a new CG matrix $\mathbf{A}'^{(t)}$ with a reduced condition number. In our second main theoretical result, we analyze $\kappa(\mathbf{A}'^{(t)})$, the bound that it induces on error, and the implications for U.

Theorem 2: Let $\mathbf{A}^{(t)} := \beta\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \text{diag}\{\boldsymbol{\alpha}^{(t)}\}$ denote the SBL inverse-covariance matrix at the t-th iteration of Alg. 1. Let $\mathbf{M}^{(t)} := \text{diag}\{\beta\boldsymbol{\theta} + \boldsymbol{\alpha}^{(t)}\}$ denote the preconditioner, where $\boldsymbol{\theta} \in \mathbb{R}^D$ is a vector of positive values. Define the preconditioned matrix $\mathbf{A}'^{(t)} := (\mathbf{M}^{(t)})^{-1/2}\mathbf{A}^{(t)}(\mathbf{M}^{(t)})^{-1/2}$. Let $\boldsymbol{b}^{(t)} \in \mathbb{R}^D$ be any vector and $\boldsymbol{b}'^{(t)} := (\mathbf{M}^{(t)})^{-1/2}\boldsymbol{b}^{(t)}$. Let $\epsilon^{(t)}$ be the relative residual error after running U conjugate gradient steps to solve the system $\mathbf{A}'^{(t)}\boldsymbol{x}'^{(t)} = \boldsymbol{b}'^{(t)}$ with $\boldsymbol{x}_0'^{(t)} = \mathbf{0}$. Given $(\mathcal{S},\mathcal{U},\hat{\boldsymbol{\alpha}})$-convergence, it follows that

$$\lim_{t\to\infty} \epsilon^{(t)} \leq 2\exp\left(-U\sqrt{\frac{1-\|\boldsymbol{\Theta}^{-1}\boldsymbol{\Phi}_\mathcal{S}^\top\boldsymbol{\Phi}_\mathcal{S} - \mathbf{I}\|_2}{1+\|\boldsymbol{\Theta}^{-1}\boldsymbol{\Phi}_\mathcal{S}^\top\boldsymbol{\Phi}_\mathcal{S} - \mathbf{I}\|_2}}\right), \tag{29}$$

where $\boldsymbol{\Theta} = \text{diag}\{\boldsymbol{\theta}_\mathcal{S}\}$ and $\mathbf{I}$ is the identity matrix.

(29) indicates that faster convergence is achieved for $\boldsymbol{\theta}_\mathcal{S} \in \mathbb{R}^{|\mathcal{S}|}$ that minimizes $\|\text{diag}\{\boldsymbol{\theta}_\mathcal{S}\}^{-1}\boldsymbol{\Phi}_\mathcal{S}^\top\boldsymbol{\Phi}_\mathcal{S} - \mathbf{I}\|_2$. In practice, the support $\mathcal{S}$ is not known in advance, so we may instead choose $\boldsymbol{\theta} \in \mathbb{R}^D$ to minimize $\|\text{diag}\{\boldsymbol{\theta}\}^{-1}\boldsymbol{\Phi}^\top\boldsymbol{\Phi} - \mathbf{I}\|_2$. For example, if $\boldsymbol{\Phi}^\top\boldsymbol{\Phi}$ is a diagonal matrix (i.e. all columns of $\boldsymbol{\Phi}$ are mutually orthogonal), then the optimal choice is $\theta_j = \sum_{i=1}^N \Phi_{i,j}^2$, which corresponds to the Jacobi preconditioner. The proof of Theorem 2 is given in Appendix B.

C. A Theory for U and K for Compressed Sensing Matrices

We now show that our bounds in Theorems 1 and 2 can be simplified for a large class of matrices $\boldsymbol{\Phi}$ satisfying the restricted isometry property (RIP) and commonly employed in compressed sensing applications.

Definition 2 (Restricted Isometry Property): Let $\boldsymbol{\Phi} \in \mathbb{R}^{N\times D}$, $d \leq D$, and $\delta > 0$. Then, $\boldsymbol{\Phi}$ satisfies (d,δ)-RIP if for every set $\mathcal{C} \subseteq \{1,\ldots,D\}$ of size $|\mathcal{C}| = d$ and every vector $\boldsymbol{v} \in \mathbb{R}^d$,

$$(1-\delta)\|\boldsymbol{v}\|_2^2 \leq \|\boldsymbol{\Phi}_\mathcal{C}\boldsymbol{v}\|_2^2 \leq (1+\delta)\|\boldsymbol{v}\|_2^2. \tag{30}$$

Corollary 2: Let $\boldsymbol{\Phi} \in \mathbb{R}^{N\times D}$ satisfy (d,δ)-RIP. For any $\mathcal{C} \subseteq \mathbb{N}_D$ with $|\mathcal{C}| = d$, let $\boldsymbol{\Delta}_{\mathcal{C},\mathcal{C}} := \boldsymbol{\Phi}_\mathcal{C}^\top\boldsymbol{\Phi}_\mathcal{C} - \mathbf{I}$. Then, $\|\boldsymbol{\Delta}_{\mathcal{C},\mathcal{C}}\|_2 \leq \delta$.

Proof: From (30), we have for any vector $\boldsymbol{v} \in \mathbb{R}^d$,

$$1-\delta \leq \frac{\boldsymbol{v}^\top\boldsymbol{\Phi}_\mathcal{C}^\top\boldsymbol{\Phi}_\mathcal{C}\boldsymbol{v}}{\boldsymbol{v}^\top\boldsymbol{v}} \leq 1+\delta \Rightarrow \left|\frac{\boldsymbol{v}^\top\boldsymbol{\Delta}_{\mathcal{C},\mathcal{C}}\boldsymbol{v}}{\boldsymbol{v}^\top\boldsymbol{v}}\right| \leq \delta. \tag{31}$$

(31) bounds the Rayleigh quotient (and eigenvalues) of $\boldsymbol{\Delta}_{\mathcal{C},\mathcal{C}}$ between $[-\delta,\delta]$. The spectral norm of a symmetric matrix is equal to its largest absolute eigenvalue. ■

Corollary 3: In Theorem 1, let $\boldsymbol{\Phi} \in \mathbb{R}^{N\times D}$ satisfy (d,δ)-RIP, where $d = |\mathcal{S}|$. Then, for $j \in \mathcal{S}$, the standard deviation of the diagonal estimator $\nu_j^{(t)}$ satisfies

$$\lim_{t\to\infty} \nu_j^{(t)} \leq \frac{1}{\sqrt{K}} \cdot \frac{\delta}{\beta(1-\delta)}. \tag{32}$$

Proof: In (25), take $\boldsymbol{\Theta} = \mathbf{I}$, which reduces the numerator to $\|\boldsymbol{\Phi}_\mathcal{S}^\top\boldsymbol{\Phi}_\mathcal{S} - \mathbf{I}\|_2$. By Corollary 2, this quantity is at most δ. Finally, by Definition 2, $\sigma_{\min}^2(\boldsymbol{\Phi}_\mathcal{S}) \geq (1-\delta)$. ■

Corollary 4: In Theorem 2, let $\boldsymbol{\Phi} \in \mathbb{R}^{N\times D}$ satisfy (d,δ)-RIP, where $d = |\mathcal{S}|$. Let $\theta_j = 1$ for all $j \in \mathbb{N}_D$. Then, the relative residual error $\epsilon^{(t)}$ of conjugate gradient satisfies

$$\lim_{t\to\infty} \epsilon^{(t)} \leq 2\exp\left(-U\sqrt{\frac{1-\delta}{1+\delta}}\right). \tag{33}$$

Proof: This follows from applying Corollary 2 to (29). ■

One may have expected that as N and D increase, CoFEM's hyperparameters U and K must also increase proportionally to ensure that ν_j and ϵ remain small. However, Corollaries 3 and 4 illustrate that for RIP matrices,[3] the bounds on U and K only depend on δ (not N or D). In compressed sensing, δ can be small even as N and D grow very large [36]. In Section V, we use this insight to demonstrate that even for very large D, CoFEM can accurately perform sparse coding with small constant values for U and K.

[2] Theoretical results for CG often define ϵ in terms of the norm weighted by $\mathbf{A}^{-1}$ (or $\mathbf{A}$), even though $\epsilon_2 := \|\boldsymbol{r}_U\|_2/\|\boldsymbol{b}\|_2$ is used to determine convergence in software. We follow this convention while noting that one can exploit the relation $\epsilon_2 \leq \epsilon\sqrt{\kappa(\mathbf{A})}$ to extend our results for ϵ_2.

[3] RIP is one mathematical notion for the idea of "close to orthonormality" for a set of dictionary columns. There exist other notions (e.g. incoherence, null space property) [36]. We conjecture there are bounds similar to Corollaries 3 and 4 that hold for matrices that satisfy these other properties.

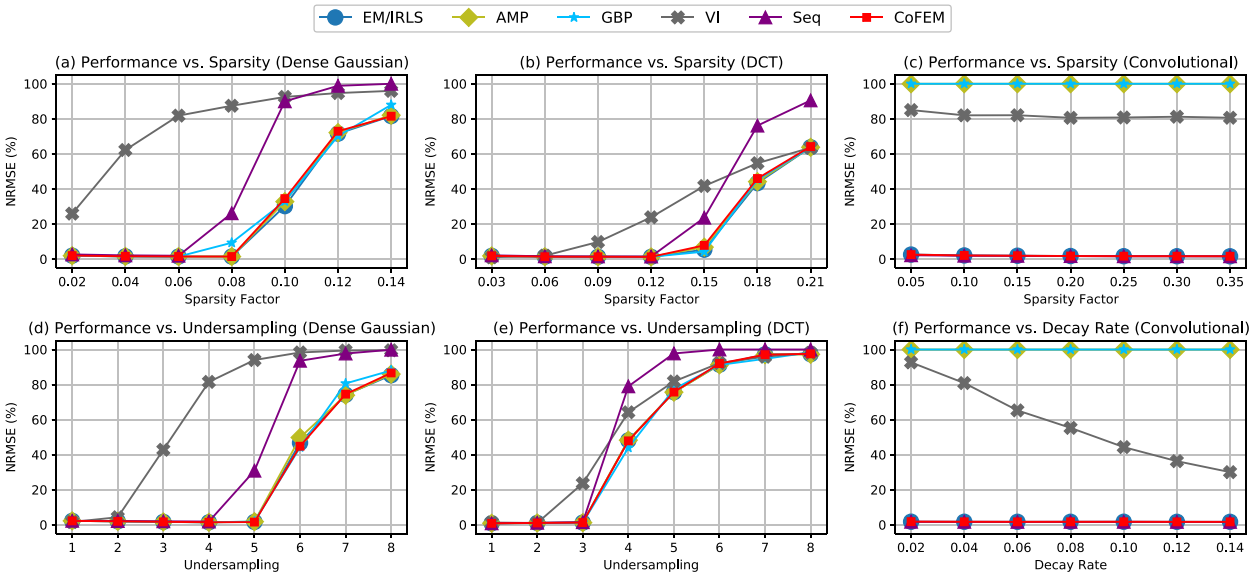


Fig. 1. Comparing the accuracy of different SBL inference schemes. Each point represents the mean of 25 trials.

V. Simulated Experiments

In this section, we run a series of experiments to compare the accuracy and scalability of CoFEM to that of other SBL inference methods across a broad range of different settings.

A. Experimental Setup

1) General Structure: We design all simulations with a structure inspired by Section V-A of [3]. We form a ground-truth latent vector $\boldsymbol{z}^* \in \mathbb{R}^D$ of spikes by drawing d of its components from a distribution $\mathcal{P}$ and setting the other $D-d$ components to zero. The location of the spikes are chosen uniformly at random. Given a dictionary $\boldsymbol{\Phi} \in \mathbb{R}^{N\times D}$, the observed data $\boldsymbol{y} \in \mathbb{R}^N$ is generated as $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{z}^* + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2\mathbf{I}) \in \mathbb{R}^N$ with $\sigma = 0.01$. The goal is to apply SBL for reconstructing $\boldsymbol{z}^*$. Success is measured through minimization of normalized root mean squared error (NRMSE), $||\hat{\boldsymbol{z}} - \boldsymbol{z}^*||_2/||\boldsymbol{z}^*||_2 \times 100$, where $\hat{\boldsymbol{z}} = \boldsymbol{\mu}$ is the mean of the distribution $p(\boldsymbol{z} \mid \boldsymbol{y}, \hat{\boldsymbol{\alpha}})$ upon convergence.

2) Dictionary: We consider three types of dictionaries $\boldsymbol{\Phi}$:

- **Dense Gaussian**: We draw each element $\Phi_{i,j}$ independently from $\mathcal{N}(0, 1/N)$ to form a dense matrix $\boldsymbol{\Phi}$. The spike distribution of $\boldsymbol{z}^*$ is $\mathcal{P} := \text{Uniform}(-2, 2)$.
- **DCT**: We let $\boldsymbol{\Phi} = \mathbf{M}\boldsymbol{\Omega}^{-1}$, where $\boldsymbol{\Omega} \in \mathbb{R}^{D\times D}$ is the matrix corresponding to the 1D discrete cosine transform of size D and $\mathbf{M} \in \mathbb{R}^{N\times D}$ is an undersampling operator that selects N out of D components (where $N \leq D$). The spike distribution is $\mathcal{P} := \mathcal{N}(0, 5)$.
- **Convolutional**: We set $N = D$ and $\boldsymbol{\Phi} \in \mathbb{R}^{D\times D}$ to a convolution in which its columns are delayed (and truncated) repetitions of an exponentially decaying filter $\boldsymbol{\phi} \in \mathbb{R}^D$. That is, $\phi_j := (1-\rho)^{j-1}$ for $0 < \rho < 1$. Thus, $\boldsymbol{\Phi}$ is a lower triangular matrix in which the j-th column is a concatenation of $j-1$ zeros and $\{\phi_1, \ldots, \phi_{N-(j-1)}\}$. The spike distribution is $\mathcal{P} := \text{Exponential}(1.5)$.

B. Accuracy Analysis

For our first analysis, we fix $D = 1{,}024$ and evaluate the accuracy of various SBL inference schemes across different settings. We compare among EM/IRLS [2], AMP [17], GBP [18], VI [19], Seq [20], and CoFEM. Note that EM and IRLS always yield the same result; IRLS is just the Woodbury identity (8) applied to EM. All EM-based methods (EM/IRLS, AMP, GBP, VI, CoFEM) are executed for $T = 50$ iterations. AMP and GBP both employ $T_{\text{amp}} = T_{\text{gbp}} = 10$ inner steps. For CoFEM, Corollary 3 tells us that we can keep the number of probes K very small since $\delta \approx 0$ and $\beta = 1/(0.01)^2 = 10{,}000$. We use $K = 20$ probe vectors, though we have found that even smaller values for K do not change the results. We employ $U = 400$ maximum CG steps with early termination if the residual error drops below the threshold $\epsilon_{\text{max}} = 10^{-4}$. Results are displayed in Fig. 1.

1) Performance Vs. Sparsity: In Fig. 1(a), we consider the dense Gaussian dictionary with $N = \lfloor D/4 \rfloor$, which is typical in a *compressed sensing* setting. Let the *sparsity factor* $f \in [0, 1]$ determine the number of non-zero coefficients in the latent signal $\boldsymbol{z}^*$ as $d = \lfloor f \cdot D \rfloor$. We vary f and observe its impact on NRMSE. At low f, all algorithms perform well except VI. As f increases, EM/IRLS, AMP, GBP, and CoFEM exhibit the same decay in performance, while Seq decays more rapidly. In Fig. 1(b), we consider the DCT dictionary with $N = \lfloor D/3 \rfloor$ measurements, again varying f. We see the same overall trend as in Fig. 1(a). In Fig. 1(c), we compare NRMSE versus sparsity level f for the convolutional dictionary. The decay rate is set to $\rho = 0.04$. EM/IRLS, Seq, and CoFEM have near-perfect NRMSE at all f. However, VI fails again due to its biased objective, while AMP and GBP both fail to converge due to the convolutional dictionary $\boldsymbol{\Phi}$.

2) Performance Vs. Undersampling: In Fig. 1(d), we revisit the Gaussian dictionary and fix $f = 0.06$. We vary the *undersampling rate* $r > 0$, where $N = \lfloor D/r \rfloor$. We see that EM/IRLS,

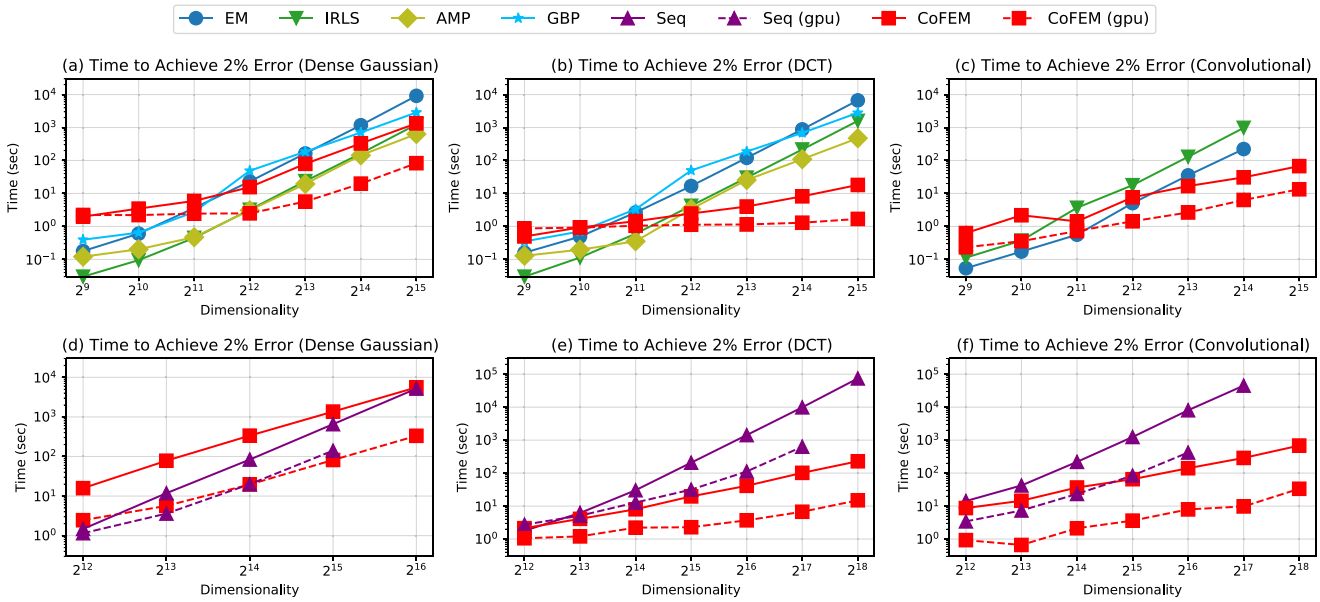


Fig. 2. Comparing the scalability of different SBL inference schemes on log-log scales. Some lines omit points for high dimensions due to memory issues.

AMP, GBP, and CoFEM have similar performance at all r, while VI and Seq degrade more rapidly with increasing r. A similar trend is shown in Fig. 1(e), in which we vary r for the DCT dictionary while fixing $f = 0.12$.

3) Performance Vs. Decay Rate: Finally, in Fig. 1(f), we consider the convolutional dictionary with fixed $f = 0.2$ and vary the decay factor ρ. Smaller ρ leads to slower decay of the exponential filter, which increases the correlations between the columns of $\mathbf{\Phi}$. We observe that EM/IRLS, Seq, and CoFEM are all robust to changes in ρ. The performance of VI is heavily correlated with ρ, while AMP and GBP diverge for all values of ρ.

CoFEM is the only algorithm that is as robust as EM/IRLS to changes in sparsity level, undersampling rate, and correlation between dictionary columns. We attribute this robustness to the fact that CoFEM performs an unbiased estimation of the posterior variances, regardless of the dictionary structure.

C. Scalability Analysis

We now analyze the scalability of CoFEM and other algorithms, by considering how computation time and memory requirements change as D is increased. In all settings, CoFEM has $K = 20$ probes and $U = 400$ maximum CG steps (with $\epsilon_{\max} = 10^{-4}$), regardless of D. The preconditioner employs $\theta_j = 1$ for all j. Results are in Fig. 2.

1) CoFEM Vs. EM-Based Algorithms: We begin by comparing CoFEM against the other EM-based algorithms (EM, IRLS, AMP, GBP). We record the time needed by each EM-based algorithm to obtain low NRMSE (i.e. $\leq 2\%$); typically at most 30 EM iterations are sufficient for the experiments we consider here. VI is omitted because it is unable to reach this threshold in many cases, as illustrated by Fig. 1.

For our first setting (Fig. 2(a)), we consider the dense Gaussian $\mathbf{\Phi}$ with $D = 2^p$ for $p = \{9, 10, \ldots, 15\}$. For each D, we let $N = \lfloor D/4 \rfloor$ and $d = \lfloor 0.06D \rfloor$. As D increases, we observe that CoFEM becomes faster than EM by an order of magnitude. CoFEM is slower than AMP, GBP, and IRLS, yet the gap decreases for large D. This is because it takes $\tau_D = O(DN)$-time to apply a dense $\mathbf{\Phi}$ to a vector, so CoFEM has the same asymptotic complexity as AMP and GBP for the dense case (see Table I). Furthermore, due to its space-saving and parallelization-friendly design, CoFEM can exploit a GPU[4] to be up to $7\times$ faster than IRLS/AMP/GBP and $100\times$ faster than EM.

Next, in Fig. 2(b) and Fig. 2(c), we repeat the experiment of increasing D for the two structured $\mathbf{\Phi}$ (DCT and convolutional). In both cases, there are fast algorithms for applying $\mathbf{\Phi}$ to an arbitrary vector in $\tau_D = O(D \log D)$-time. As a result, CoFEM is faster at high D than all other algorithms. In the DCT case, we let $d = \lfloor 0.12D \rfloor$ and $N = \lfloor D/3 \rfloor$ for all D. We observe that CoFEM can be faster than EM by up to $360\times$ on the CPU and $3800\times$ on the GPU, reducing over two hours of computation for EM at $D = 2^{15}$ to two seconds. In the convolutional case, we let $d = \lfloor 0.2D \rfloor$. For $D = 2^{15}$, EM and IRLS have memory issues due to the large $N = D$. AMP and GBP are unable to yield sensible solutions (see Fig. 1(c)). In contrast, CoFEM is accurate while being faster than EM/IRLS and not experiencing memory issues at high D.

2) CoFEM Vs. Seq: Finally, we compare CoFEM to the sequential algorithm. Both of these algorithms have low space complexity (see Table I), preventing memory issues at very high D. We repeat the experimental settings from Fig. 2(a)–(c) at higher dimensions $D = 2^p$ for $p = \{12, 13, \ldots, 18\}$.

For the dense dictionary (Fig. 2(d),[5] we observe that Seq is faster than CoFEM on the CPU for low D, but the gap decreases

[4]We use a 16-GB Nvidia T4 GPU and a 32-GB, 2.3 GHz Intel Xeon CPU.

[5]We could not run $D > 2^{16}$ for the dense case because there is not enough memory on our devices to store all the entries of $\mathbf{\Phi}$.

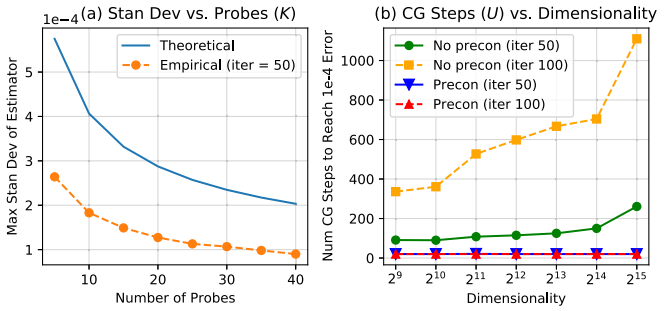


Fig. 3. Empirical insights into U and K based on theory of CoFEM.

for high D. On the GPU, CoFEM is faster than Seq at moderate D due to its superior ability to exploit parallelized hardware. For the two structured dictionaries (Fig. 2(e) and 2(f)), CoFEM is faster across all settings. We further observe that for many of the larger dimensions, Seq inevitably encounters memory issues as a covariance-based method. In contrast, CoFEM has no such issue due to its low space complexity. For example, at $D = 2^{18}$, CoFEM can leverage the GPU to be $5000\times$ faster than Seq. In summary, CoFEM's ability to obviate covariance computation provides many advantages in terms of scalability over existing SBL inference schemes.

We emphasize that across all of our experiments, U and K are kept at small values despite substantially increasing D, demonstrating CoFEM's scalability at very high dimensions.

D. Empirical Insights From the Theory of CoFEM

Finally, we give some insights into our theory for CoFEM's hyperparameters U and K (Section IV). We use the DCT dictionary setup of Section V-A with $N = \lfloor D/3 \rfloor$ measurements.

In Fig. 3(a), we let $D = 1{,}024$ and plot the relationship between (1) the number of probes K and (2) the maximum standard deviation $\max_{j=1}^{D} \nu_j^{(T)}$ over all D coordinates of the diagonal estimator at iteration $T = 50$ of CoFEM. We present a "theoretical" curve calculated using Theorem 1 with $\boldsymbol{\Theta} = \mathbf{I}$ and an "empirical" curve in which each $\nu_j^{(T)}$ is estimated through the empirical standard deviation of 1000 Monte Carlo trials. Fig. 3(a) shows that our theoretical bound (1) captures the true decay of the standard deviation as a function of K and (2) accurately upperbounds the empirical curve after T *finite* iterations despite Theorem 1's condition of $T \to \infty$.

In Fig. 3(b), we consider different orders of magnitude for D and plot the number of CG steps U needed to achieve small error $\epsilon_{\text{max}} = 10^{-4}$ at the $T = 50$-th and $T = 100$-th iterations of CoFEM. We explore the impact of our diagonal preconditioner (Section III-C) on U. The figure reveals that our theoretical insights from Section IV-B and Theorem 2 hold in practice: (1) when there is no preconditioner and ϵ_{max} is fixed, U needs to grow substantially with increasing iterations T and/or dimensionality D, yet (2) when a preconditioner is used, U can be small and constant for large T and large D.

VI. Real-Data Experiments

We now demonstrate the utility of CoFEM for two real data applications – calcium deconvolution and MRI reconstruction.

A. Calcium Deconvolution

Calcium imaging is a widely used tool in neuroscience for monitoring the electrical activity of neurons [37]. It is a method for indirectly observing the spiking activity of a neuron through a fluorescence trace $\boldsymbol{y} \in \mathbb{R}^D$, approximated as the convolution of an intrinsically sparse spiking pattern $\boldsymbol{z}^* \in \mathbb{R}^D$ with a decaying calcium response $\boldsymbol{\phi} \in \mathbb{R}^D$. In calcium deconvolution, we aim to recover $\boldsymbol{z}^*$ from $\boldsymbol{y}$ and $\boldsymbol{\phi}$.

1) SBL Model: We can cast calcium deconvolution as a SBL problem with a dictionary $\boldsymbol{\Phi} \in \mathbb{R}^{D\times D}$ consisting of delayed (and truncated) versions of $\boldsymbol{\phi}$ as its columns, similar to the setting in Section V-A2. The data is then assumed to be generated as $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{z}^* + \boldsymbol{\varepsilon}$, where each $\varepsilon_i \sim \mathcal{N}(0, 1/\beta)$. Since $\boldsymbol{z}^*$ for calcium deconvolution is a non-negative vector, we employ SBL with *non-negativity constraints* (Section III-E2).

2) Spike Inference: We use the CoFEM inference algorithm adapted for non-negativity constraints. Since $\boldsymbol{\Phi}$ represents a discrete-time convolution, we can efficiently apply $\boldsymbol{\Phi}$ to any vector $\boldsymbol{v}$ through fast Fourier transforms and an element-wise product. Non-negative SBL yields a rectified Gaussian posterior $p(\boldsymbol{z} \mid \boldsymbol{y}, \hat{\boldsymbol{\alpha}})$ over the latent spikes $\boldsymbol{z}$. To obtain a point estimate $\hat{\boldsymbol{z}}$, we find a *filtered mode*.[6] Specifically, we first filter $\boldsymbol{z}$ by selecting components z_j that are highly likely to be non-zero, i.e. z_j such that $p(z_j = 0 \mid \boldsymbol{y}, \hat{\boldsymbol{\alpha}}) < q$, where q is some small percentile (e.g. 0.05, 0.01). This query is possible only because SBL models uncertainty in $\boldsymbol{z}$. Setting the unselected components of $\boldsymbol{z}$ to zero, we then find the most likely values for all selected z_j's according to the posterior, resulting in $\hat{\boldsymbol{z}}$. More details can be found in Appendix C-1. This is analogous to thresholding heuristics commonly used by ℓ_1-based sparse coding algorithms [38]. However, unlike those value-based strategies, the percentile filtering for SBL is value-agnostic and instead operates on the learned posterior.

3) Data and Hyperparameters: We apply SBL to five fluorescence traces from the GENIE dataset [39]. Each $\boldsymbol{y}$ contains data at a sampling rate of $\nu = 60$ Hz for a total of $D = 14{,}400$ time points, which is a high-dimensional problem. The response $\boldsymbol{\phi}$ has $\phi_i = (1-\psi)^{i-1}$ for $\psi = 1/(\nu \times 0.7) = 0.0238$, a widely-used value for the calcium indicator GCaMP6f. CoFEM employs $T = 20$, $K = 20$ and $U = 400$. The noise precision β is estimated from $\boldsymbol{y}$ through a Fourier domain procedure, as described in [40]. To obtain $\hat{\boldsymbol{z}}$, we use a filter percentile of $q = 0.05$.

4) Results: To evaluate $\hat{\boldsymbol{z}}$, we employ the following standard practice [40]: The GENIE dataset has ground-truth times for neural spikes. Let $\boldsymbol{z}^* \in \mathbb{R}^D$ be a zero-one vector indicating when true spiking occurred. For bin length b, we reduce $\boldsymbol{z}^*$ and $\hat{\boldsymbol{z}}$ to vectors $\boldsymbol{c}^*$ and $\hat{\boldsymbol{c}}$ of length $\lceil D/b \rceil$ by summing across windows of b consecutive components. We then compute the Pearson

[6] The location parameter $\boldsymbol{\mu}$ is a poor point estimate, since $\boldsymbol{\mu}$ is not equal to the mode due to the asymmetry of the *rectified* Gaussian distribution.

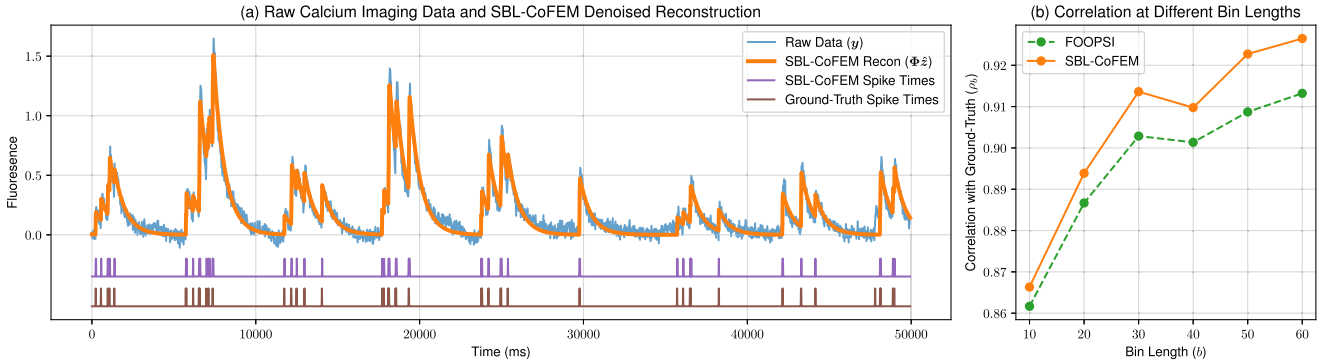


Fig. 4. Results of running SBL-CoFEM for calcium imaging and comparison with FOOPSI.

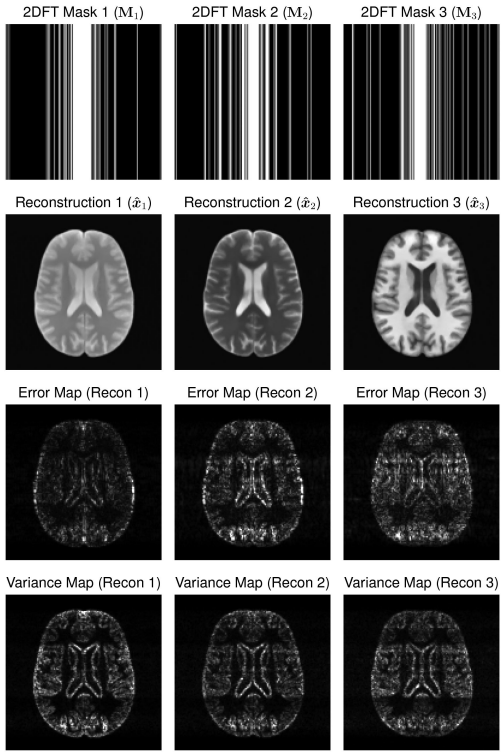


Fig. 5. Undersampling k-space masks and SBL-CoFEM reconstrutions for the SRI24 atlas. Error maps are scaled by 15× to aid visualization.

correlation coefficient ρ_b between $\boldsymbol{c}^*$ and $\hat{\boldsymbol{c}}$. A high value for ρ_b indicates agreement between $\hat{\boldsymbol{z}}$ and $\boldsymbol{z}^*$.

Fig. 4(a) plots sample SBL-CoFEM outputs and compares its inferred spike times with the ground truth. Fig. 4(b) shows an averaged curve over the five traces of ρ_b versus b at various bin lengths $b \in \{10, 20, 30, 40, 50, 60\}$ for SBL-CoFEM and a popular ℓ_1-based method called FOOPSI [41]. The figure shows that SBL-CoFEM outperforms FOOPSI, with the gap growing for larger bin sizes b.

B. Multi-Contrast MRI Reconstruction

Magnetic resonance imaging (MRI) is one of the dominant modalities for imaging the human body [42]. The standard data acquisition practice samples a set of points (called "k-space") $\boldsymbol{k} \in \mathbb{C}^N$ from the two-dimensional Fourier transform (2DFT) of the image $\boldsymbol{x} \in \mathbb{R}^D$.[7] In practice, one may aim to collect $N < D$ points to reduce the amount of time a patient needs to remain in the scanner. However, doing so leads to an ill-posed inverse problem $\mathbf{M}\mathbf{F}\boldsymbol{x} = \boldsymbol{k}$ for $\boldsymbol{x}$, where $\mathbf{F} \in \mathbb{C}^{D\times D}$ is the 2DFT and $\mathbf{M} \in \mathbb{R}^{N\times D}$ is an undersampling operator. Thus, compressed sensing strategies often exploit the sparsity of $\boldsymbol{x}$ with respect to some transform for accurate reconstruction.

In *multi-contrast* MRI reconstruction, there are L images $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L$ of an object that one wishes to recover from corresponding undersampled k-space measurements $\boldsymbol{k}_1, \ldots, \boldsymbol{k}_L$. Bilgic *et al.* [6] showed that SBL with *multi-task learning* (Section III-E1) achieves successful joint recovery of the multiple images. They outperformed ℓ_1-based methods by exploiting common sparsity patterns among the horizontal/vertical image gradients (i.e. row-wise/column-wise finite differences) of $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_L$. However, the main drawback is the computation time for reconstruction, which is many times slower than ℓ_1 methods. We demonstrate how CoFEM can accelerate this method while maintaining its superior performance.

1) SBL Model and Inference: For each contrast ℓ, let $\boldsymbol{\Phi}_\ell = \mathbf{M}_\ell\mathbf{F}$ denote the ℓ-th undersampled 2DFT operator with undersampling mask $\mathbf{M}_\ell$. The task is to infer the sparse latent vector $\boldsymbol{z}_\ell^{\text{horz}} = \partial^{\text{horz}}\boldsymbol{x}_\ell \in \mathbb{R}^D$, where ∂^{horz} denotes the horizontal image gradient operator. Note that ∂^{horz} is a convolution, implying that it corresponds to a diagonal matrix $\boldsymbol{\Delta}^{\text{horz}}$ in the Fourier domain. Let the observed data be $\boldsymbol{y}_\ell^{\text{horz}} = \mathbf{M}_\ell\boldsymbol{\Delta}^{\text{horz}}\mathbf{M}_\ell^\top\boldsymbol{k}_\ell$ for all ℓ. We impose a multi-task SBL model on $(\boldsymbol{z}_\ell^{\text{horz}}, \boldsymbol{y}_\ell^{\text{horz}}, \boldsymbol{\Phi}_\ell)$, as per (16). We use a shared $\boldsymbol{\alpha}^{\text{horz}}$ to ensure that we will learn grouped sparsity patterns among $\{\boldsymbol{z}_\ell^{\text{horz}}\}_{\ell=1}^L$. The same procedure is repeated for $\boldsymbol{z}_\ell^{\text{vert}}$ based on the operator ∂^{vert}. Further details are in [6].

We employ CoFEM for multi-task SBL (Section III-E1) to recover $p(\boldsymbol{z}_\ell^{\text{horz}} \mid \boldsymbol{y}_\ell^{\text{horz}}, \hat{\boldsymbol{\alpha}}^{\text{horz}})$ and $p(\boldsymbol{z}_\ell^{\text{vert}} \mid \boldsymbol{y}_\ell^{\text{vert}}, \hat{\boldsymbol{\alpha}}^{\text{vert}})$ for all ℓ. The reconstructed image $\hat{\boldsymbol{x}}_\ell$ are computed from these posterior distributions. More details are given in Appendix D-1.

2) Data and Hyperparameters: We consider the SRI24 atlas [43], a set of $L = 3$ MRI contrasts with dimensions 200×200 for a total of $D = 40{,}000$ pixels. For each image $\boldsymbol{x}_\ell^*$, we undersample its 2DFT by a factor of four in the horizontal

[7] Though MRIs are generally complex-valued, the data we use here are real-valued. See [6] for how to generalize SBL to the complex case.

TABLE II
RESULTS ON MULTI-CONTRAST MRI RECONSTRUCTION

Algorithm	NRMSE	Computation Time
SparseMRI	5.4%	22.3 min
SBL-Seq	3.4%	89.9 min
SBL-CoFEM	2.9%	2.5 min (CPU), 0.2 min (GPU)

dimension, observing $N = 10{,}000$ points to form $\boldsymbol{k}_\ell$. The mask $\mathbf{M}_\ell$ is randomly determined according to a power rule favoring the center of k-space [44]. For CoFEM, we have $T = 15$, $\beta = 10^6$, $K = 8$, $\epsilon_{\text{max}} = 10^{-5}$ and $U = 200$. The algorithm is not sensitive to variations in these values.

3) Results: Fig. 5 provides images of the masks $\mathbf{M}_\ell$ and SBL-CoFEM's reconstructions $\hat{\boldsymbol{x}}_\ell$. Success is measured through NRMSE between the vectorized forms of $\hat{\boldsymbol{x}} \in \mathbb{R}^{D\cdot L}$ and $\boldsymbol{x}^* \in \mathbb{R}^{D\cdot L}$. Table II compares SBL-CoFEM against SparseMRI (ℓ_1-based compressed sensing [44]) and SBL-Seq (SBL with sequential algorithm [6]). Although SBL-Seq has lower NRMSE than SparseMRI, it requires high computation time. In contrast, SBL-CoFEM attains the lowest error and can exploit the GPU to be 450× faster than SBL-Seq.

Finally, the bottom half of Fig. 5 displays error maps of absolute differences between $\hat{\boldsymbol{x}}_\ell$ and $\boldsymbol{x}^*_\ell$, along with *variance maps* for each image. Each variance map captures the model's confidence over different areas of its reconstruction; pixels with high variance indicate more potential to deviate from the point estimate $\hat{\boldsymbol{x}}_\ell$. SBL can create variance maps because it models uncertainty; non-Bayesian methods that do not model uncertainty (e.g. ℓ_1 methods) cannot generate these maps. Appendix D-2 explains how SBL-CoFEM can generate these variance maps using the diagonal estimation rule. The variance maps bear similarity to the ground-truth error maps, suggesting that SBL-CoFEM can predict its errors in reconstruction.

VII. CONCLUSION

We developed covariance-free expectation-maximization (CoFEM) to accelerate sparse Bayesian learning (SBL). By solving linear systems to circumvent matrix inversion, CoFEM exhibits superior time-efficiency and space-efficiency over existing SBL inference schemes, especially when the dictionary $\boldsymbol{\Phi}$ admits fast matrix-vector multiplication. We theoretically analyzed CoFEM's hyperparameters, such as the number of linear systems and number of solver steps, showing that they can remain small even at high dimensions. Coupled with GPU acceleration, CoFEM can be up to thousands of times faster than other SBL methods without sacrificing sparse coding accuracy. Finally, we used CoFEM for real-data applications, showing that it can adapt to multi-task learning and non-negativity constraints, while enabling SBL to be competitive with non-Bayesian methods in accuracy and scalability.

APPENDIX A
PROOF OF THEOREM 1

Proof: We begin by characterizing $\hat{\boldsymbol{\Sigma}} := \lim_{t\to\infty} \boldsymbol{\Sigma}^{(t)}$. By the block matrix inversion formula [45], any symmetric positive definite matrix with diagonal blocks $\mathbf{X}, \mathbf{Z}$ and off-diagonal block $\mathbf{Y}$ and can be inverted as

$$\begin{bmatrix} \mathbf{X} & \mathbf{Y} \\ \mathbf{Y}^\top & \mathbf{Z} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{X}^{-1} + \mathbf{X}^{-1}\mathbf{Y}\mathbf{W}\mathbf{Y}^\top\mathbf{X}^{-1} & -\mathbf{X}^{-1}\mathbf{Y}\mathbf{W} \\ -\mathbf{W}\mathbf{Y}^\top\mathbf{X}^{-1} & \mathbf{W} \end{bmatrix}, \tag{34}$$

where $\mathbf{W} := (\mathbf{Z} - \mathbf{Y}^\top\mathbf{X}^{-1}\mathbf{Y})^{-1}$ is the inverse Schur complement. We apply (26) to $\boldsymbol{\Sigma}^{(t)}$, with $\mathbf{X} := \beta\boldsymbol{\Phi}_\mathcal{S}^\top\boldsymbol{\Phi}_\mathcal{S} + \text{diag}\{\boldsymbol{\alpha}_\mathcal{S}^{(t)}\}$, $\mathbf{Y} := \beta\boldsymbol{\Phi}_\mathcal{S}^\top\boldsymbol{\Phi}_\mathcal{U}$, and $\mathbf{Z} := \beta\boldsymbol{\Phi}_\mathcal{U}^\top\boldsymbol{\Phi}_\mathcal{U} + \text{diag}\{\boldsymbol{\alpha}_\mathcal{U}^{(t)}\}$. As $t \to \infty$, we have $\alpha_j^{(t)} \to \infty$ for $j \in \mathcal{U}$, which forces $\mathbf{W} \to \mathbf{0}$, where $\mathbf{0}$ is the zero-matrix. Then, by (26),

$$\hat{\boldsymbol{\Sigma}} = \lim_{t\to\infty} \boldsymbol{\Sigma}^{(t)} = \begin{bmatrix} (\beta\boldsymbol{\Phi}_\mathcal{S}^\top\boldsymbol{\Phi}_\mathcal{S} + \text{diag}\{\hat{\boldsymbol{\alpha}}_\mathcal{S}\})^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{35}$$

(24) follows from applying (35) to Lemma 1 for $j \in \mathcal{U}$; since all rows of $\hat{\boldsymbol{\Sigma}}$ that correspond to $\mathcal{U}$ are zero, the estimator's standard deviation must also converge to zero.

We now prove (25) for an active index $j \in \mathcal{S}$. Let $\psi_j > 0$ be any positive real number. Using Lemma 1 and (35), we can bound $\hat{\nu}_j := \lim_{t\to\infty} \nu_j^{(t)}$ with

$$\hat{\nu}_j = \sqrt{\frac{1}{K}\sum\nolimits_{j'\in\mathcal{T}_j} \hat{\Sigma}_{j,j'}^2} \le \frac{1}{\sqrt{K}}\sqrt{(\hat{\Sigma}_{j,j} - \psi_j)^2 + \sum\nolimits_{j'\in\mathcal{T}_j} \hat{\Sigma}_{j,j'}^2}, \tag{36}$$

where $\mathcal{T}_j := \mathcal{S} \setminus \{j\}$. Let $\boldsymbol{\Psi} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ be a diagonal matrix with ψ_j for all $j \in \mathcal{S}$ along its diagonal. Let $\boldsymbol{e}_j$ be the standard unit vector in $\mathbb{R}^{|\mathcal{S}|}$ corresponding to j. From (36), we have

$$\begin{aligned} \hat{\nu}_j &\le \tfrac{1}{\sqrt{K}}\|(\hat{\boldsymbol{\Sigma}}_{\mathcal{S},\mathcal{S}} - \boldsymbol{\Psi})\boldsymbol{e}_j\|_2 \\ &\le \tfrac{1}{\sqrt{K}}\|\hat{\boldsymbol{\Sigma}}_{\mathcal{S},\mathcal{S}} - \boldsymbol{\Psi}\|_2 = \tfrac{1}{\sqrt{K}}\|\boldsymbol{\Psi}(\hat{\boldsymbol{\Sigma}}_{\mathcal{S},\mathcal{S}}^{-1} - \boldsymbol{\Psi}^{-1})\hat{\boldsymbol{\Sigma}}_{\mathcal{S},\mathcal{S}}\|_2 \\ &\le \tfrac{1}{\sqrt{K}}\|\boldsymbol{\Psi}(\hat{\boldsymbol{\Sigma}}_{\mathcal{S},\mathcal{S}}^{-1} - \boldsymbol{\Psi}^{-1})\|_2\|\hat{\boldsymbol{\Sigma}}_{\mathcal{S},\mathcal{S}}\|_2 \\ &= \tfrac{1}{\sqrt{K}}\|\boldsymbol{\Psi}(\beta\boldsymbol{\Phi}_\mathcal{S}^\top\boldsymbol{\Phi}_\mathcal{S} + \text{diag}\{\hat{\boldsymbol{\alpha}}_\mathcal{S}\} - \boldsymbol{\Psi}^{-1})\|_2\|\hat{\boldsymbol{\Sigma}}_{\mathcal{S},\mathcal{S}}\|_2, \end{aligned} \tag{37}$$

where the last step uses (35) to expand $\hat{\boldsymbol{\Sigma}}_{\mathcal{S},\mathcal{S}}^{-1}$.

We now perform a change-of-variables: Let $\boldsymbol{\Theta} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ be any diagonal matrix of positive diagonal values. We define $\boldsymbol{\Psi} := (\beta\boldsymbol{\Theta} + \text{diag}\{\hat{\boldsymbol{\alpha}}_\mathcal{S}\})^{-1}$ and re-write (37) as

$$\begin{aligned} \hat{\nu}_j &\le \tfrac{1}{\sqrt{K}}\|(\beta\boldsymbol{\Theta} + \text{diag}\{\hat{\boldsymbol{\alpha}}_\mathcal{S}\})^{-1}(\beta\boldsymbol{\Phi}_\mathcal{S}^\top\boldsymbol{\Phi}_\mathcal{S} - \beta\boldsymbol{\Theta})\|_2\|\hat{\boldsymbol{\Sigma}}_{\mathcal{S},\mathcal{S}}\|_2 \\ &\le \tfrac{1}{\sqrt{K}}\|(\beta\boldsymbol{\Theta})^{-1}(\beta\boldsymbol{\Phi}_\mathcal{S}^\top\boldsymbol{\Phi}_\mathcal{S} - \beta\boldsymbol{\Theta})\|_2\|\hat{\boldsymbol{\Sigma}}_{\mathcal{S},\mathcal{S}}\|_2 \\ &= \tfrac{1}{\sqrt{K}}\|\boldsymbol{\Theta}^{-1}\boldsymbol{\Phi}_\mathcal{S}^\top\boldsymbol{\Phi}_\mathcal{S} - \mathbf{I}\|_2\|\hat{\boldsymbol{\Sigma}}_{\mathcal{S},\mathcal{S}}\|_2. \end{aligned} \tag{38}$$

Finally, we can bound the last term of (38) with

$$\|\hat{\boldsymbol{\Sigma}}_{\mathcal{S},\mathcal{S}}\|_2 = \frac{1}{\lambda_{\min}(\hat{\boldsymbol{\Sigma}}_{\mathcal{S},\mathcal{S}}^{-1})} \le \frac{1}{\beta\cdot\lambda_{\min}(\boldsymbol{\Phi}_\mathcal{S}^\top\boldsymbol{\Phi}_\mathcal{S})}, \tag{39}$$

where we use the fact that the singular values coincide with the eigenvalues for a symmetric positive definite matrix. ■

APPENDIX B
PROOF OF THEOREM 2

Proof: The statement follows from (27) and showing that $\hat{\kappa} := \lim_{t\to\infty} \kappa(\mathbf{A}'^{(t)}) \leq (1+\xi)/(1-\xi)$, where $\xi := \|\boldsymbol{\Theta}^{-1}\boldsymbol{\Phi}_{\mathcal{S}}^{\top}\boldsymbol{\Phi}_{\mathcal{S}} - \mathbf{I}\|_2$. Let $\boldsymbol{\Delta} := \beta\boldsymbol{\Phi}^{\top}\boldsymbol{\Phi} - \text{diag}\{\beta\boldsymbol{\theta}\}$. Then,

$$\mathbf{A}^{(t)} = \mathbf{M}^{(t)} + \boldsymbol{\Delta} \Rightarrow \mathbf{A}'^{(t)} = \mathbf{I} + (\mathbf{M}^{(t)})^{-\frac{1}{2}}\boldsymbol{\Delta}(\mathbf{M}^{(t)})^{-\frac{1}{2}}$$

$$\Rightarrow \hat{\mathbf{A}}' := \lim_{t\to\infty} \mathbf{A}'^{(t)} = \mathbf{I} + \begin{bmatrix} \hat{\mathbf{M}}_{\mathcal{S},\mathcal{S}}^{-\frac{1}{2}}\boldsymbol{\Delta}_{\mathcal{S},\mathcal{S}}\hat{\mathbf{M}}_{\mathcal{S},\mathcal{S}}^{-\frac{1}{2}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad (40)$$

where $\hat{\mathbf{M}} := \text{diag}\{\beta\boldsymbol{\theta} + \hat{\boldsymbol{\alpha}}\}$.

Our goal is to bound $\hat{\kappa} = \lambda_{\max}(\hat{\mathbf{A}}')/\lambda_{\min}(\hat{\mathbf{A}}')$. Equation (40) shows that if η is an eigenvalue of $\hat{\mathbf{M}}_{\mathcal{S},\mathcal{S}}^{-1/2}\boldsymbol{\Delta}_{\mathcal{S},\mathcal{S}}\hat{\mathbf{M}}_{\mathcal{S},\mathcal{S}}^{-1/2}$, then $1+\eta$ is an eigenvalue of $\hat{\mathbf{A}}'$. This reduces our task to bounding η. A matrix $\mathbf{X} \in \mathbb{R}^{D\times D}$ is similar to another matrix $\mathbf{Y} \in \mathbb{R}^{D\times D}$ if there exists an invertible matrix $\mathbf{Z} \in \mathbb{R}^{D\times D}$ such that $\mathbf{Y} = \mathbf{Z}^{-1}\mathbf{X}\mathbf{Z}$, and this further implies that $\mathbf{X}$ and $\mathbf{Y}$ have the same eigenvalues [45]. Since $\mathbf{X} := \hat{\mathbf{M}}_{\mathcal{S},\mathcal{S}}^{-1/2}\boldsymbol{\Delta}_{\mathcal{S},\mathcal{S}}\hat{\mathbf{M}}_{\mathcal{S},\mathcal{S}}^{-1/2}$ is similar to $\mathbf{Y} := \hat{\mathbf{M}}_{\mathcal{S},\mathcal{S}}^{-1}\boldsymbol{\Delta}_{\mathcal{S},\mathcal{S}}$ for $\mathbf{Z} := \hat{\mathbf{M}}_{\mathcal{S},\mathcal{S}}^{1/2}$, it follows that η is also an eigenvalue of $\mathbf{Y}$. The absolute eigenvalues of a matrix cannot exceed its spectral norm, implying

$$|\eta| \leq \|\hat{\mathbf{M}}_{\mathcal{S},\mathcal{S}}^{-1}\boldsymbol{\Delta}_{\mathcal{S},\mathcal{S}}\|_2 = \|\hat{\mathbf{M}}_{\mathcal{S},\mathcal{S}}^{-1}(\beta\boldsymbol{\Phi}_{\mathcal{S}}^{\top}\boldsymbol{\Phi}_{\mathcal{S}} - \beta\boldsymbol{\Theta})\|_2$$

$$\leq \|(\beta\boldsymbol{\Theta})^{-1}(\beta\boldsymbol{\Phi}_{\mathcal{S}}^{\top}\boldsymbol{\Phi}_{\mathcal{S}} - \beta\boldsymbol{\Theta})\|_2 = \|\boldsymbol{\Theta}^{-1}\boldsymbol{\Phi}_{\mathcal{S}}^{\top}\boldsymbol{\Phi}_{\mathcal{S}} - \mathbf{I}\|_2. \quad (41)$$

It follows that $\lambda_{\max}(\hat{\mathbf{A}}') \leq 1 + \|\boldsymbol{\Theta}^{-1}\boldsymbol{\Phi}_{\mathcal{S}}^{\top}\boldsymbol{\Phi}_{\mathcal{S}} - \mathbf{I}\|_2$ and $\lambda_{\min}(\hat{\mathbf{A}}') \geq 1 - \|\boldsymbol{\Theta}^{-1}\boldsymbol{\Phi}_{\mathcal{S}}^{\top}\boldsymbol{\Phi}_{\mathcal{S}} - \mathbf{I}\|_2$, which bounds $\kappa(\hat{\mathbf{A}}')$. ■

APPENDIX C
DETAILS OF SBL FOR CALCIUM DECONVOLUTION

1) Filtered Mode: Let $\mathcal{S} \subseteq \{1, 2, \ldots, D\}$ be the set of selected indices after percentile filtering of $p(\boldsymbol{z} \mid \boldsymbol{y}, \hat{\boldsymbol{\alpha}})$ recovered by CoFEM. Let $\boldsymbol{\Phi}_{\mathcal{S}} \in \mathbb{R}^{N\times|\mathcal{S}|}$ denote the sub-matrix of $\boldsymbol{\Phi}$ composed of the columns corresponding to $\mathcal{S}$. Then, the *filtered mode* is the solution to the following problem:

$$\hat{\boldsymbol{u}} = \arg\min_{\boldsymbol{u}\geq\mathbf{0}\in\mathbb{R}^{|\mathcal{S}|}} ||\boldsymbol{y} - \boldsymbol{\Phi}_{\mathcal{S}}\boldsymbol{u}||_2^2 + \sum_{j\in\mathcal{S}}(\hat{\alpha}_j/\beta)u_j^2, \quad (42)$$

which can be obtained using non-negative least-squares solvers. Solving (42) is fast in practice, because it is a low-dimensional problem (i.e. $|\mathcal{S}| \ll D$). Our point estimate solution is $\hat{\boldsymbol{z}}$, where $\hat{z}_j = u_j$ for $j \in \mathcal{S}$ and $\hat{z}_j = 0$ for $j \notin \mathcal{S}$.

APPENDIX D
DETAILS OF SBL FOR MRI RECONSTRUCTION

1) Final Reconstruction: Let $\boldsymbol{\mu}_{\ell}^{\text{horz}}$ and $\boldsymbol{\mu}_{\ell}^{\text{vert}}$ denote the respective means of $p(\boldsymbol{z}_{\ell}^{\text{horz}} \mid \boldsymbol{y}_{\ell}^{\text{horz}}, \hat{\boldsymbol{\alpha}}^{\text{horz}})$ and $p(\boldsymbol{z}_{\ell}^{\text{vert}} \mid \boldsymbol{y}_{\ell}^{\text{vert}}, \hat{\boldsymbol{\alpha}}^{\text{vert}})$. These quantities are combined through solving a constrained least-squares problem to yield a final reconstruction $\hat{\boldsymbol{x}}_{\ell}$:

$$\hat{\boldsymbol{x}}_{\ell} = \arg\min_{\boldsymbol{x}_{\ell}} \|\partial^{\text{horz}}\boldsymbol{x}_{\ell} - \boldsymbol{\mu}_{\ell}^{\text{horz}}\|_2^2 + \|\partial^{\text{vert}}\boldsymbol{x}_{\ell} - \boldsymbol{\mu}_{\ell}^{\text{vert}}\|_2^2,$$

$$\text{s.t. } \mathbf{M}_{\ell}\mathbf{F}\boldsymbol{x}_{\ell} = \boldsymbol{k}_{\ell}. \quad (43)$$

We use Parseval's Theorem [32] to cast (43) to the Fourier domain. This converts ∂^{horz} and ∂^{vert} into diagonal matrices $\boldsymbol{\Delta}^{\text{horz}}$ and $\boldsymbol{\Delta}^{\text{vert}}$ in the Fourier domain, giving an element-wise separable problem with a closed-form solution [6].

2) Variance Map: For each MRI contrast ℓ, SBL learns posterior distributions $\mathcal{N}(\boldsymbol{\mu}_{\ell}^{\text{horz}}, \boldsymbol{\Sigma}_{\ell}^{\text{horz}})$ and $\mathcal{N}(\boldsymbol{\mu}_{\ell}^{\text{vert}}, \boldsymbol{\Sigma}_{\ell}^{\text{vert}})$ for the image gradients. We use the fact that for $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and a matrix $\mathbf{E}$, we have $\mathbf{E}\boldsymbol{z} \sim \mathcal{N}(\mathbf{E}\boldsymbol{\mu}, \mathbf{E}\boldsymbol{\Sigma}\mathbf{E}^{\top})$. The solution to (43) has the form $\hat{\boldsymbol{x}}_{\ell} = \mathbf{E}_1\boldsymbol{\mu}_{\ell}^{\text{horz}} + \mathbf{E}_2\boldsymbol{\mu}_{\ell}^{\text{horz}} + \boldsymbol{e}$ for some matrices $\mathbf{E}_1, \mathbf{E}_2 \in \mathbb{R}^{\text{D}\times\text{D}}$ and some vector $\boldsymbol{e} \in \mathbb{R}^D$. To obtain variance maps for $\hat{\boldsymbol{x}}_{\ell}$, we find the diagonal elements of its covariance matrix $\boldsymbol{\Psi} := \mathbf{E}_1\boldsymbol{\Sigma}_{\ell}^{\text{horz}}\mathbf{E}_1^{\top} + \mathbf{E}_2\boldsymbol{\Sigma}_{\ell}^{\text{vert}}\mathbf{E}_2^{\top}$. We can do this by drawing probes and applying the diagonal estimation rule (Section III-A). In doing so, we need to apply $\boldsymbol{\Sigma}_{\ell}^{\text{horz}} = (\beta\boldsymbol{\Phi}^{\top}\boldsymbol{\Phi} + \text{diag}\{\hat{\boldsymbol{\alpha}}^{\text{horz}}\})^{-1}$ and $\boldsymbol{\Sigma}_{\ell}^{\text{vert}} = (\beta\boldsymbol{\Phi}^{\top}\boldsymbol{\Phi} + \hat{\boldsymbol{\alpha}}^{\text{vert}})^{-1}$ to arbitrary vectors, which can be done via parallel CG (Section III-B).

REFERENCES

[1] D. J. MacKay, "Bayesian methods for backpropagation networks," in *Models of Neural Networks III*. New York, NY, USA: Springer, 1996, pp. 211–254.

[2] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, 2001.

[3] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2346–2356, Jun. 2008.

[4] S. Ji, D. Dunson, and L. Carin, "Multitask compressive sensing," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 92–106, Jan. 2009.

[5] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2153–2164, Aug. 2004.

[6] B. Bilgic, V. K. Goyal, and E. Adalsteinsson, "Multi-contrast reconstruction with Bayesian compressed sensing," *Magn. Reson. Med.*, vol. 66, no. 6, pp. 1601–1615, 2011.

[7] S. Liu, J. Jia, Y. D. Zhang, and Y. Yang, "Image reconstruction in electrical impedance tomography based on structure-aware sparse Bayesian learning," *IEEE Trans. Med. Imag.*, vol. 37, no. 9, pp. 2090–2102, Sep. 2018.

[8] P. Chen, Z. Cao, Z. Chen, and X. Wang, "Off-grid DOA estimation using sparse Bayesian learning in MIMO radar with unknown mutual coupling," *IEEE Trans. Signal Process.*, vol. 67, no. 1, pp. 208–220, Jan. 2019.

[9] Y. Ma, Y. Konishi, K. Kinoshita, S. Lao, and M. Kawade, "Sparse Bayesian regression for head pose estimation," in *Proc. 18th Int. Conf. Pattern Recognit.*, 2006, pp. 507–510.

[10] Z. Zhang, T.-P. Jung, S. Makeig, and B. D. Rao, "Compressed sensing for energy-efficient wireless telemonitoring of noninvasive fetal ECG via block sparse Bayesian learning," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 2, pp. 300–309, Feb. 2013.

[11] S. Yuan, S. Wang, M. Ma, Y. Ji, and L. Deng, "Sparse Bayesian learning-based time-variant deconvolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 11, pp. 6182–6194, Nov. 2017.

[12] O. Williams, A. Blake, and R. Cipolla, "Sparse Bayesian learning for efficient visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1292–1304, Aug. 2005.

[13] Q. Wu, Y. D. Zhang, M. G. Amin, and B. Himed, "Space-time adaptive processing and motion parameter estimation in multistatic passive radar using sparse Bayesian learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 2, pp. 944–957, Feb. 2016.

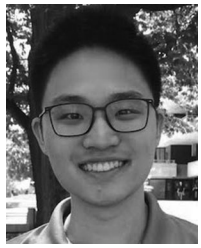
[14] J. Fang, Y. Shen, H. Li, and P. Wang, "Pattern-coupled sparse Bayesian learning for recovery of block-sparse signals," *IEEE Trans. Signal Process.*, vol. 63, no. 2, pp. 360–372, Jan. 2015.

[15] D. P. Wipf, S. S. Nagarajan, J. Platt, D. Koller, and Y. Singer, "A new view of automatic relevance determination," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 1625–1632.

[16] D. Wipf and S. Nagarajan, "Iterative reweighted $\ell_1$and ℓ_2 methods for finding sparse solutions," *IEEE J. Sel. Top. Signal Process.*, vol. 4, no. 2, pp. 317–329, Apr. 2010.

[17] J. Fang, L. Zhang, and H. Li, "Two-dimensional pattern-coupled sparse Bayesian learning via generalized approximate message passing," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2920–2930, Jun. 2016.

[18] X. Tan and J. Li, "Computationally efficient sparse Bayesian learning via belief propagation," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2010–2021, Apr. 2010.
[19] H. Duan, L. Yang, J. Fang, and H. Li, "Fast inverse-free sparse Bayesian learning via relaxed evidence lower bound maximization," *IEEE Signal Process. Lett.*, vol. 24, no. 6, pp. 774–778, Jun. 2017.
[20] M. E. Tipping and A. Faul, "Fast marginal likelihood maximisation for sparse Bayesian models," in *Proc. Int. Workshop Artif. Intell. Statist.*, 2003, pp. 276–283.
[21] D. P. Wipf, B. D. Rao, and S. Nagarajan, "Latent variable Bayesian models for promoting sparsity," *IEEE Trans. Inf. Theory*, vol. 57, no. 9, pp. 6236–6255, Sep. 2011.
[22] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Roy. Stat. Society: Ser. B. (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.
[23] Q. Su and Y.-C. Wu, "Convergence analysis of the variance in gaussian belief propagation," *IEEE Trans. Signal Process.*, vol. 62, no. 19, pp. 5119–5131, Oct. 2014.
[24] M. Al-Shoukairi, P. Schniter, and B. D. Rao, "A GAMP-based low complexity sparse Bayesian learning algorithm," *IEEE Trans. Signal Process.*, vol. 66, no. 2, pp. 294–308, Jan. 2018.
[25] C. M. Bishop and M. Tipping, "Variational relevance vector machines," in *Proc. Uncertainty Artif. Intell.*, pp. 46–53, 2000.
[26] S. D. Babacan, M. Luessi, R. Molina, and A. K. Katsaggelos, "Low-rank matrix completion by variational sparse Bayesian learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2011, pp. 2188–2191.
[27] D. Shutin, T. Buchgraber, S. R. Kulkarni, and H. V. Poor, "Fast variational sparse Bayesian learning with automatic relevance determination for superimposed signals," *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 6257–6261, Dec. 2011.
[28] B. Worley, "Scalable mean-field sparse Bayesian learning," *IEEE Trans. Signal Process.*, vol. 67, no. 24, pp. 6314–6326, Dec. 2019.
[29] C. Bekas, E. Kokiopoulou, and Y. Saad, "An estimator for the diagonal of a matrix," *Appl. Numer. Math.*, vol. 57, no. 11–12, pp. 1214–1229, 2007.
[30] M. R. Hestenes *et al.*, "Methods of conjugate gradients for solving linear systems," *J. Res. Nat. Bur. Standards*, vol. 49, no. 6, pp. 409–436, Dec. 1952.
[31] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," Tech. Rep., pp. 1–64, 1994.
[32] A. V. Oppenheim, J. R. Buck, and R. W. Schafer, *Discrete-Time Signal Processing*, vol. 2. Upper Saddle River, NJ, USA: Prentice Hall, 2001.
[33] D. P. Wipf and B. D. Rao, "An empirical Bayesian strategy for solving the simultaneous sparse approximation problem," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3704–3716, Jul. 2007.
[34] Q. Wu, Y. D. Zhang, M. G. Amin, and B. Himed, "Complex multitask Bayesian compressive sensing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2014, pp. 3375–3379.
[35] A. Nalci, I. Fedorov, M. Al-Shoukairi, T. T. Liu, and B. D. Rao, "Rectified Gaussian scale mixtures and the sparse non-negative least squares problem," *IEEE Trans. Signal Process.*, vol. 66, no. 12, pp. 3124–3139, Jun. 2018.
[36] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*, 2013.
[37] C. Grienberger and A. Konnerth, "Imaging calcium in neurons," *Neuron*, vol. 73, no. 5, pp. 862–885, 2012.
[38] J. Friedrich, P. Zhou, and L. Paninski, "Fast online deconvolution of calcium imaging data," *PLoS Comput. Biol.*, vol. 13, no. 3, 2017, Art. no. e1005423.
[39] J. Akerboom *et al.*, "Optimization of a gcamp calcium indicator for neural activity imaging," *J. Neurosci.*, vol. 32, no. 40, pp. 13819–13840, 2012.
[40] E. A. Pnevmatikakis *et al.*, "Simultaneous denoising, deconvolution, and demixing of calcium imaging data," *Neuron*, vol. 89, no. 2, pp. 285–299, 2016.
[41] J. T. Vogelstein *et al.*, "Fast nonnegative deconvolution for spike train inference from population calcium imaging," *J. Neuriophysiol.*, vol. 104, no. 6, pp. 3691–3704, 2010.
[42] D. G. Nishimura, *Principles of Magnetic Resonance Imaging*. Stanford, CA, USA: Standford Univ., 2010.
[43] T. Rohlfing, N. M. Zahr, E. V. Sullivan, and A. Pfefferbaum, "The sri24 multichannel atlas of normal adult human brain structure," *Hum. Brain Mapping*, vol. 31, no. 5, pp. 798–819, 2010.
[44] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magn. Reson. Med.: An Official J. Int. Soc. Magn. Reson. Med.*, vol. 58, no. 6, pp. 1182–1195, 2007.
[45] G. Strang, *Linear Algebra and Its Applications*. Belmont, CA: Thomson, Brooks/Cole, 2006.



Alexander Lin (Student Member, IEEE) received the A.B. degree in computer science and the S.M. degree in computational science and engineering in 2019 from Harvard University, Cambridge, MA, USA, where he is currently working toward the Ph.D. degree in computer science with the School of Engineering and Applied Sciences, advised by Professor Demba Ba. From 2019 to 2020, he was a Research Engineer with the Natural Language Processing Group, ASAPP, New York, NY, USA. In 2022, he was a Summer Research Intern with the Biomedical Machine Learning Group, Microsoft Research (New England), Cambridge, MA, USA. His research interests include machine learning, signal processing, Bayesian statistics, and biomedical imaging. In 2022, he was the recipient of the National Defense Science and Engineering Graduate Fellowship.

Andrew H. Song (Student Member, IEEE) received the B.Sc., M.Eng., and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2015, 2016, and 2022, respectively. His research interests include statistical/biological signal processing, with a focus on dictionary learning, and the connection between biological signal processing and deep learning. He was a Member of Neuroscience Statistics Research Laboratory, MIT and Computation, Representations, and Inference in Signal Processing group, Harvard University, Cambridge, MA, USA.

Berkin Bilgic received the B.S. degrees in electrical and electronics engineering and physics from Bogazici University, Istanbul, Turkey, in 2008, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA. He was a Postdoctoral Researcher with the Martinos Center for Biomedical Imaging and became an Instructor in radiology with Massachusetts General Hospital (MGH), Boston, MA, USA and Harvard Medical School (HMS), Boston, MA, USA, in 2016. Since 2018, he has been an Affiliated Faculty with the Harvard-MIT Department of Health Sciences and Technology and an Assistant Professor in Radiology with MGH/HMS since 2019. His group develops data acquisition and image reconstruction techniques for efficient magnetic resonance imaging.

Demba Ba (Member, IEEE) received the B.Sc. degree in electrical engineering from the University of Maryland, College Park, MD, USA, in 2004, and the M.Sci. and Ph.D. degrees in electrical engineering and computer science with a minor in mathematics from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2006 and 2011, respectively. In 2006 and 2009, he was a Summer Research Intern with Communication and Collaboration Systems Group, Microsoft Research, Redmond, WA, USA. From 2011 to 2014, he was a Postdoctoral Associate with the MIT/Harvard Neuroscience Statistics Research Laboratory, where he developed theory and efficient algorithms to assess synchrony among large assemblies of neurons. He is currently an Associate Professor of electrical engineering and bioengineering with Harvard University, Cambridge, MA, USA, where he directs the CRISP Group. His research interests include the intersection of high-dimensional statistics, optimization and dynamic modeling, with applications to neuroscience and multimedia signal processing, and the connection between neural networks, sparse signal processing, and hierarchical representations of sensory signals in the brain. In 2016, he was the recipient of a Research Fellowship in Neuroscience from the Alfred P. Sloan Foundation.