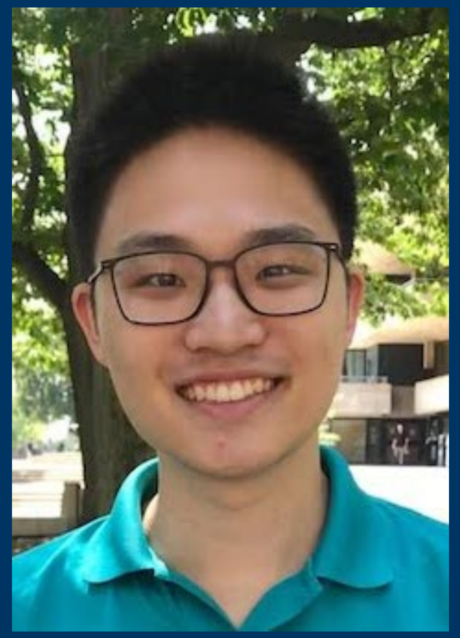
High-Dimensional Sparse Bayesian Learning without Covariance Matrices

IEEE ICASSP 2022 Paper #4913



Alexander Lin

Harvard University
School of Engineering

Andrew H. Song

Harvard Medical School
Brigham and Women's Hospital

Berkin Bilgic

Harvard Medical School
Martinos Center

Demba Ba

Harvard University
School of Engineering

Introduction

Known Dictionary $\in \mathbb{R}^{N \times D}$

Observed Data $\in \mathbb{R}^N$

Unknown Sparse Signal $\in \mathbb{R}^D$

How to solve $y = \Phi z +$ noise, where z is sparse?

- Many applications: sparse regression, compressed sensing, linear inverse problems
- Option #1: Matching pursuit/L0 methods: $\min |z|_0$
- Option #2: Basis pursuit/LASSO/L1 methods: $\min |z|_1$
- Option #3: **Sparse Bayesian Learning (SBL)**
  - Also known as: automatic relevance determination, relevance vector machine, Bayesian compressed sensing
  - Provides uncertainty quantification, automatic tuning, favorable optimization properties

[1] Wipf, D., & Nagarajan, S. (2007). A new view of automatic relevance determination.

Our Work

- Problem: SBL is slow!
  - $O(D^3)$-time and $O(D^2)$-space
  - Impractical for high dimensions
- Main Contribution: New algorithm to make SBL much faster at high D
  - $O(\tau_D)$-time and $O(D)$-space, where τ_D is the time needed to multiply Φ by a vector
  - Up to thousands of times faster than existing algorithms in practice

Sparse Bayesian Learning (SBL)

Model Overview

Objective

Solve $y = \Phi z$ + noise, where z is sparse

SBL Model

Prior $z \sim \mathcal{N}(0, \text{diag}(\alpha)^{-1}), \quad \alpha \in \mathbb{R}_+^D$

Likelihood $y \mid z \sim \mathcal{N}(\Phi z, 1/\beta \cdot \mathrm{I})$

↓

Posterior $z \mid y \sim \mathcal{N}(\mu, \Sigma)$

$$\mu = \beta \Sigma \Phi^\top y \qquad \Sigma = (\beta \Phi^\top \Phi + \text{diag}(\alpha))^{-1}$$

How to Achieve Posterior Sparsity

Optimize marginal likelihood [1, 2]:

$$\max_\alpha \log p(y; \alpha) = \max_\alpha \log \int p(y \mid z) p(z; \alpha) dz$$

Then some $\alpha_j \to \infty$,

- Then prior $p(z_j) \to$ point mass at 0
- Then posterior $p(z_j \mid y) \to$ point mass at 0

[1] MacKay, D. J. (1996). Bayesian methods for backpropagation networks
[2] Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine.

Sparse Bayesian Learning (SBL)

Inference Algorithm

EM Inference

How to optimize $\max_{\alpha} \log p(y; \alpha)$?

Use Expectation-Maximization (EM) algorithm [1, 2]:

E-Step Given $\alpha^{(t)}$, compute posterior $p(z \mid y; \alpha^{(t)})$

$$\mu^{(t)} = \beta \Sigma^{(t)} \Phi^\top y$$

$$\Sigma^{(t)} = (\beta \Phi^\top \Phi + \text{diag}(\alpha^{(t)}))^{-1}$$

M-Step Given $p(z \mid y; \alpha^{(t)})$, update $\alpha^{(t+1)}$

$$\alpha_j^{(t+1)} = \frac{1}{\mathbb{E}[z_j^2; \alpha^{(t)}]} = \frac{1}{(\mu_j^{(t)})^2 + \Sigma_{j,j}^{(t)}}$$

[1] MacKay, D. J. (1996). Bayesian methods for backpropagation networks
[2] Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine.

Sparse Bayesian Learning (SBL)

Inference Algorithm

EM Inference

How to optimize $\max_{\alpha} \log p(y; \alpha)$?

Use Expectation-Maximization (EM) algorithm [1, 2]:

E-Step Given $\alpha^{(t)}$, compute posterior $p(z \mid y; \alpha^{(t)})$

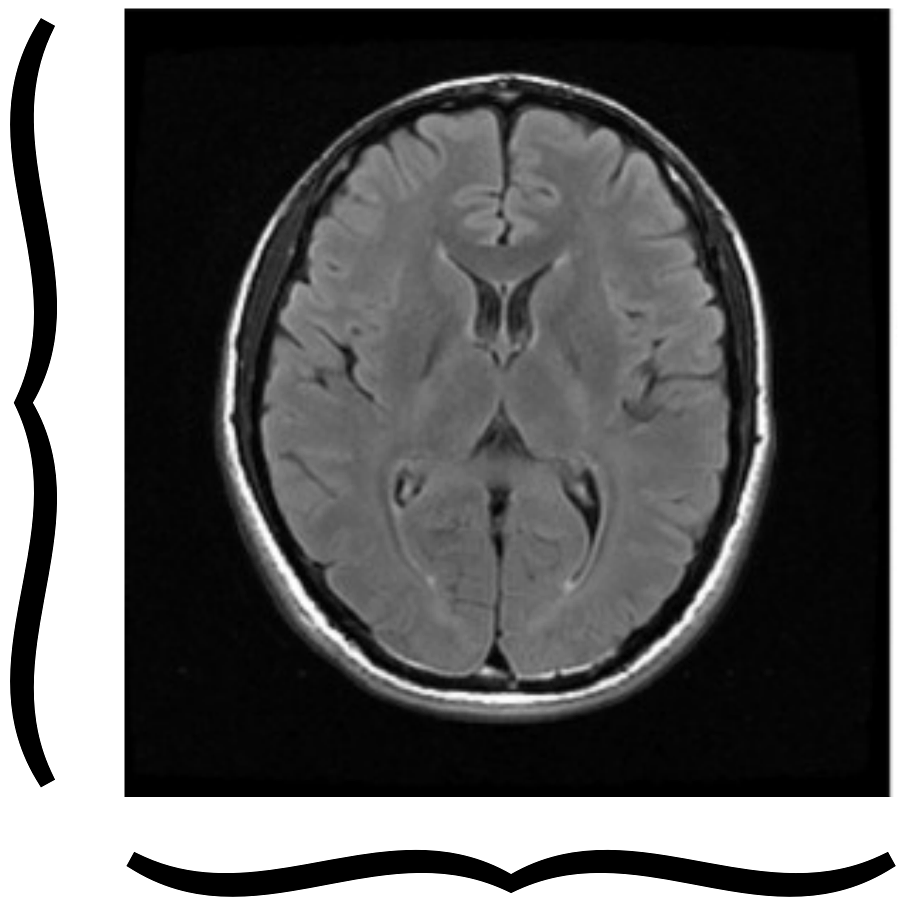
$$\mu^{(t)} = \beta \Sigma^{(t)} \Phi^{\top} y$$

$$\Sigma^{(t)} = (\beta \Phi^{\top} \Phi + \text{diag}(\alpha^{(t)}))^{-1}$$

M-Step Given $p(z \mid y; \alpha^{(t)})$, update $\alpha^{(t+1)}$

$$\alpha_j^{(t+1)} = \frac{1}{\mathbb{E}[z_j^2; \alpha^{(t)}]} = \frac{1}{(\mu_j^{(t)})^2 + \Sigma_{j,j}^{(t)}}$$

Expensive!
$O(D^3)$-time
$O(D^2)$-space



$D = 512 \cdot 512 \approx 250{,}000$

$D^2 \approx 62.5 \times 10^9$ → 250 GB to store $\Sigma^{(t)}$

$D^3 \approx 1.6 \times 10^{16}$ → ≈ 30 hours to run SBL on real MRI data [3]

Price of uncertainty is too high!

[1] MacKay, D. J. (1996). Bayesian methods for backpropagation networks
[2] Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine.
[3] Bilgic, B., Goyal, V. K., & Adalsteinsson, E. (2011). Multi-contrast reconstruction with Bayesian compressed sensing.

Sparse Bayesian Learning (SBL)

Our Contribution: Faster/Cheaper Inference via CoFEM

EM Inference

How to optimize $\max_{\alpha} \log p(y; \alpha)$?

Can use Expectation-Maximization (EM) algorithm:

E-Step Given $\alpha^{(t)}$, compute posterior $p(z \mid y; \alpha^{(t)})$

$$\mu^{(t)} = \beta \Sigma^{(t)} \Phi^\top y$$

$$\Sigma^{(t)} = (\beta \Phi^\top \Phi + \text{diag}(\alpha^{(t)}))^{-1}$$

M-Step Given $p(z \mid y; \alpha^{(t)})$, update $\alpha^{(t+1)}$

$$\alpha_j^{(t+1)} = \frac{1}{\mathbb{E}[z_j^2; \alpha^{(t)}]} = \frac{1}{\underbrace{(\mu_j^{(t)})^2}_{1} + \underbrace{\Sigma_{j,j}^{(t)}}_{2}}$$

Covariance-Free EM (CoFEM) Inference

How can we do EM without computing $\Sigma^{(t)}$?

1 Computing means $\mu_j^{(t)}$ for all $j = 1,2,\ldots,D$

$$(\Sigma^{(t)})^{-1} \mu^{(t)} = \beta \Phi^\top y$$

$$\underbrace{(\beta \Phi^\top \Phi + \text{diag}(\alpha^{(t)}))}_{A} \underbrace{\mu^{(t)}}_{x} = \underbrace{\beta \Phi^\top y}_{b}$$

This is a linear system — get x with linear solver

- Don't need physical matrix A
- Only need to compute Av for any $v \in \mathbb{R}^D$

[1] A. Lin, A. H. Song, B. Bilgic, and D. Ba. Covariance-free sparse Bayesian learning. In submission to *Transactions on Signal Processing.*
[2] A. Lin, A. H. Song, B. Bilgic, and D. Ba. High-dimensional sparse Bayesian learning without covariance matrices. *ICASSP 2022.*

Sparse Bayesian Learning (SBL)

Our Contribution: Faster/Cheaper Inference via CoFEM

EM Inference

How to optimize $\max_{\alpha} \log p(y; \alpha)$?

Can use Expectation-Maximization (EM) algorithm:

E-Step Given $\alpha^{(t)}$, compute posterior $p(z \mid y; \alpha^{(t)})$

$$\mu^{(t)} = \beta \Sigma^{(t)} \Phi^\top y$$

$$\Sigma^{(t)} = (\beta \Phi^\top \Phi + \text{diag}(\alpha^{(t)}))^{-1}$$

M-Step Given $p(z \mid y; \alpha^{(t)})$, update $\alpha^{(t+1)}$

$$\alpha_j^{(t+1)} = \frac{1}{\mathbb{E}[z_j^2; \alpha^{(t)}]} = \frac{1}{(\mu_j^{(t)})^2 + \Sigma_{j,j}^{(t)}}$$

1 2

Covariance-Free EM (CoFEM) Inference

How can we do EM without computing $\Sigma^{(t)}$?

2 Computing variances $\Sigma_{j,j}^{(t)}$ for all $j = 1, 2, \ldots, D$

Diagonal Estimation Rule [1]

Let p be a probe vector with $p_j \sim \begin{cases} +1, & \textit{prob} = 0.5 \\ -1, & \textit{prob} = 0.5 \end{cases}$

Then, for any square matrix Σ, the vector $s := p \odot \Sigma p$ is an unbiased estimator of the diagonal elements of Σ.

i.e. $\mathbb{E}[s_j] = \Sigma_{j,j}$ for all $j = 1, \ldots, D$

[1] Bekas, C., Kokiopoulou, E., & Saad, Y. (2007). An estimator for the diagonal of a matrix.

Sparse Bayesian Learning (SBL)

Our Contribution: Faster/Cheaper Inference via CoFEM

EM Inference

How to optimize $\max_{\alpha} \log p(y; \alpha)$?

Can use Expectation-Maximization (EM) algorithm:

E-Step Given $\alpha^{(t)}$, compute posterior $p(z \mid y; \alpha^{(t)})$

$$\mu^{(t)} = \beta \Sigma^{(t)} \Phi^\top y$$

$$\Sigma^{(t)} = (\beta \Phi^\top \Phi + \text{diag}(\alpha^{(t)}))^{-1}$$

M-Step Given $p(z \mid y; \alpha^{(t)})$, update $\alpha^{(t+1)}$

$$\alpha_j^{(t+1)} = \frac{1}{\mathbb{E}[z_j^2; \alpha^{(t)}]} = \frac{1}{(\mu_j^{(t)})^2 + \Sigma_{j,j}^{(t)}}$$

1 2

Covariance-Free EM (CoFEM) Inference

How can we do EM without computing $\Sigma^{(t)}$?

2 Computing variances $\Sigma_{j,j}^{(t)}$ for all $j = 1, 2, \ldots, D$

Apply Diagonal Estimation Rule:

- Compute $s = p \odot \Sigma^{(t)} p$, where p is a probe
- How to compute $\Sigma^{(t)} p$?

$$x = \Sigma^{(t)} p$$

$$(\Sigma^{(t)})^{-1} x = p$$

$$(\beta \Phi^\top \Phi + \text{diag}(\alpha^{(t)})) x = p$$

Another linear system!

Solve for x, then compute $s = p \odot x$

Sparse Bayesian Learning (SBL)

Our Contribution: Faster/Cheaper Inference via CoFEM

EM Inference

How to optimize $\max_\alpha \log p(y; \alpha)$?

Can use Expectation-Maximization (EM) algorithm:

E-Step Given $\alpha^{(t)}$, compute posterior $p(z \mid y; \alpha^{(t)})$

$$\mu^{(t)} = \beta \Sigma^{(t)} \Phi^\top y$$

$$\Sigma^{(t)} = (\beta \Phi^\top \Phi + \text{diag}(\alpha^{(t)}))^{-1}$$

M-Step Given $p(z \mid y; \alpha^{(t)})$, update $\alpha^{(t+1)}$

$$\alpha_j^{(t+1)} = \frac{1}{\mathbb{E}[z_j^2; \alpha^{(t)}]} = \frac{1}{(\mu_j^{(t)})^2 + \Sigma_{j,j}^{(t)}}$$

Covariance-Free EM (CoFEM) Inference

E-Step

- Draw a probe vector p
- Given $\alpha^{(t)}$, solve 2 linear systems in parallel:

$$\text{AX} = \text{B}$$

Inputs
$$\begin{cases} \text{A} := \beta \Phi^\top \Phi + \text{diag}(\alpha^{(t)}) : \mathbb{R}^D \to \mathbb{R}^D \\ \text{B} := [\beta \Phi^\top y \mid p] \in \mathbb{R}^{D \times 2} \end{cases}$$

Output
$$\text{X} := [\mu^{(t)} \mid x] \in \mathbb{R}^{D \times 2}$$

- Diagonal estimator $s = p \odot x$
  (high variance?)

M-Step Update $\alpha_j^{(t+1)} = \dfrac{1}{(\mu_j^{(t)})^2 + s_j^{(t)}}$

Take-away: No need to compute $\Sigma^{(t)}$!

Sparse Bayesian Learning (SBL)

Our Contribution: Faster/Cheaper Inference via CoFEM

EM Inference

How to optimize $\max_{\alpha} \log p(y; \alpha)$?

Can use Expectation-Maximization (EM) algorithm:

E-Step Given $\alpha^{(t)}$, compute posterior $p(z \mid y; \alpha^{(t)})$

$$\mu^{(t)} = \beta \Sigma^{(t)} \Phi^\top y$$

$$\Sigma^{(t)} = (\beta \Phi^\top \Phi + \text{diag}(\alpha^{(t)}))^{-1}$$

M-Step Given $p(z \mid y; \alpha^{(t)})$, update $\alpha^{(t+1)}$

$$\alpha_j^{(t+1)} = \frac{1}{\mathbb{E}[z_j^2; \alpha^{(t)}]} = \frac{1}{(\mu_j^{(t)})^2 + \Sigma_{j,j}^{(t)}}$$

Covariance-Free EM (CoFEM) Inference

E-Step

- Draw K probe vectors $p^{\langle 1 \rangle}, \ldots, p^{\langle K \rangle}$
- Given $\alpha^{(t)}$, solve $(K+1)$ linear systems in parallel:

$$\mathrm{AX} = \mathrm{B}$$

Inputs

$$\begin{cases} \mathrm{A} := \beta \Phi^\top \Phi + \text{diag}(\alpha^{(t)}) : \mathbb{R}^D \to \mathbb{R}^D \\ \mathrm{B} := [\beta \Phi^\top y \mid p^{\langle 1 \rangle} \mid \ldots \mid p^{\langle K \rangle}] \in \mathbb{R}^{D \times (K+1)} \end{cases}$$

Output

$$\mathrm{X} := [\mu^{(t)} \mid x^{\langle 1 \rangle} \mid \ldots \mid x^{\langle K \rangle}] \in \mathbb{R}^{D \times (K+1)}$$

- Diagonal estimator $s = \frac{1}{K} \sum_{k=1}^{K} p^{\langle k \rangle} \odot x^{\langle k \rangle}$ (reduced variance)

M-Step Update $\alpha^{(t+1)} = \frac{1}{(\mu_j^{(t)})^2 + s_j^{(t)}}$

Theoretical Analysis of CoFEM

	Time Complexity (per iteration)	**Space Complexity**
EM	$O(D^3)$	$O(D^2)$
CoFEM (ours)	$O(\tau_D UK)$	$O(DK)$

Satisfied by Compressed Sensing Matrices

Lin et al. [1], (Thm 1 & 2, *Informal*)

For Φ satisfying δ-RIP, ϵ (resp. ν) is a function of U (resp. K) and δ

- Implication: U and K can be kept small and constant even if D grows very large

[1] Lin, A., Song, A. H., Bilgic, B., & Ba, D. (2021). Covariance-Free Sparse Bayesian Learning.

τ_D: Time for matrix-vector multiply (MVM) $\Phi^\top \Phi v$, where $v \in \mathbb{R}^D$

- Worst case: $O(D^2)$ = dense matrix multiplication
- If Φ is **structured**: Can be much faster & matrix-free
  - Wavelet transform: $O(D)$
  - Fourier transform: $O(D \log D)$
  - Convolution: $\min\{O(Df), O(D \log D)\}$
  - Discrete cosine transform: $O(D \log D)$
  - Undersampling: $O(D)$
  - Sparse matrix multiplication: $O(S)$

Everywhere in signal processing applications!

U: Number of iterative steps inside linear solver

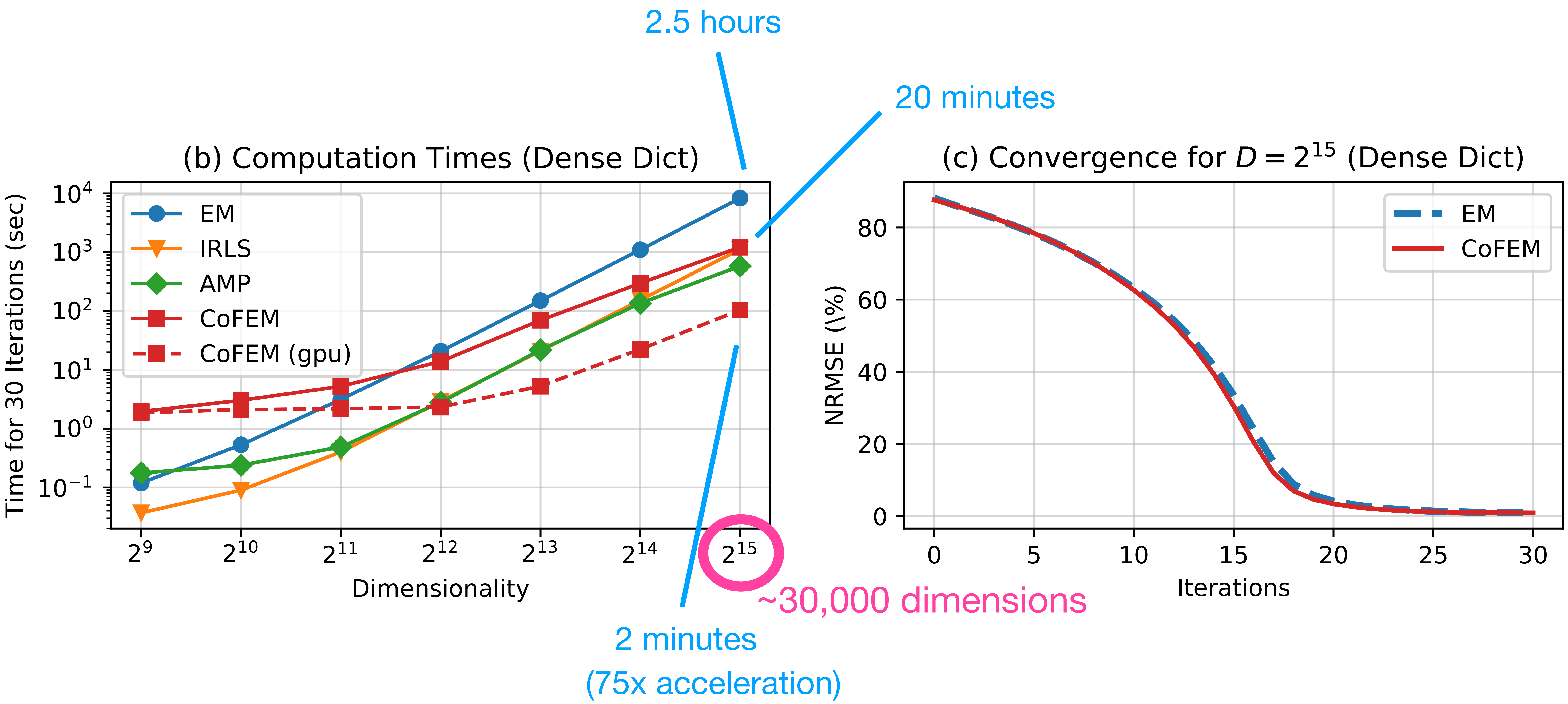
- We use *conjugate gradient* algorithm
- Larger U $\implies$ Smaller solver error ϵ

K: Number of probe vectors (parallelizable)

- Larger K $\implies$ Smaller estimator variance ν

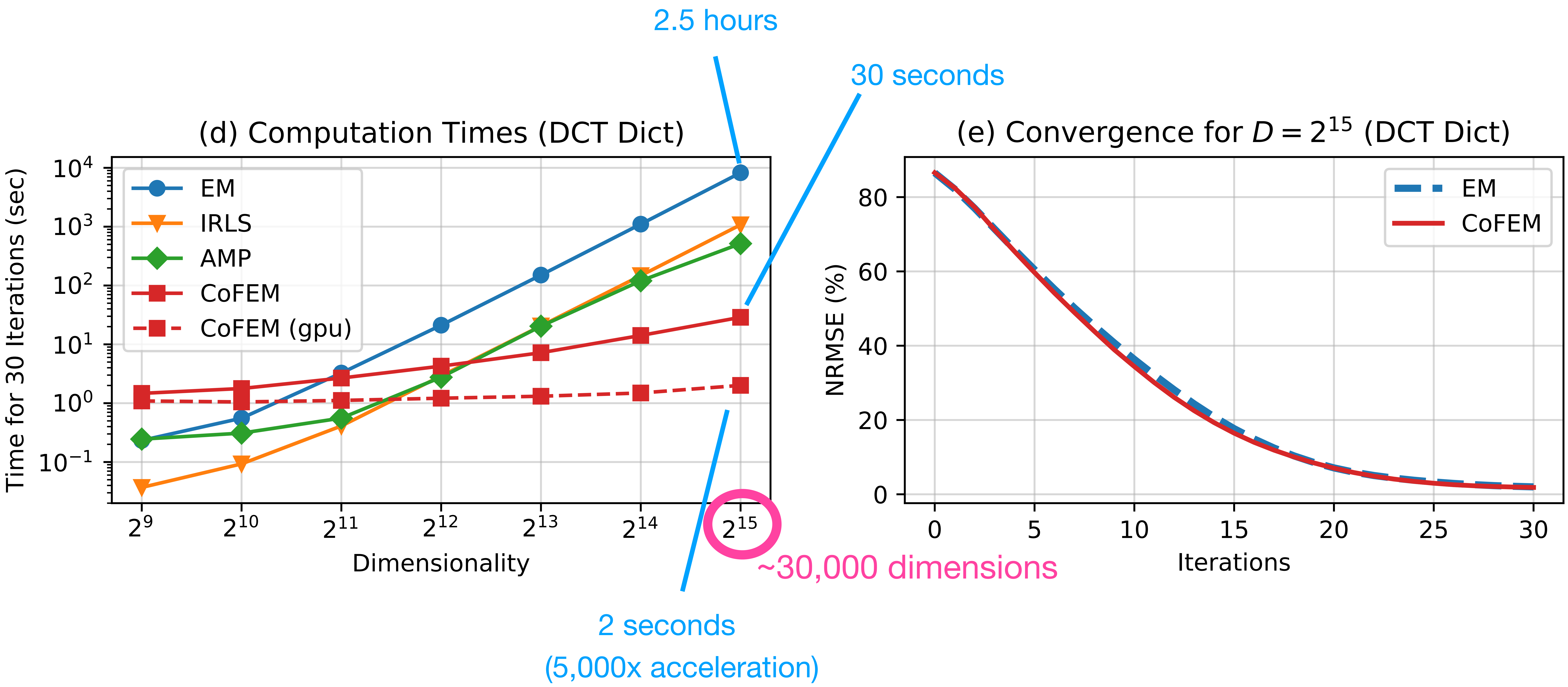
Experimental Analysis of CoFEM

Dense Forward Model Φ



Experimental Analysis of CoFEM

Structured (DCT) Forward Model Φ



For more information…

- Check out our conference paper: Lin, A., Song, A. H., Bilgic, B., & Ba, D. (2022, May). **High-Dimensional Sparse Bayesian Learning without Covariance Matrices.** In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1511-1515). IEEE.
- Check out our other works on CoFEM (can be found at https://sites.google.com/view/alexanderlin)
  - **Journal paper:** Lin, A., Song, A. H., Bilgic, B., & Ba, D. (2022). **Covariance-Free Sparse Bayesian Learning.** *arXiv preprint arXiv:2105.10439*.
    - Theorems/proofs for CoFEM, preconditioning for CoFEM's conjugate gradient algorithm, extending CoFEM to multi-task learning and non-negativity constraints, more simulated & real data experiments
  - **Applications to real MRI data**
    - Lin, A., Bilgic, B., & Ba, D. (2021). Accelerating Bayesian Compressed Sensing for Fast Multi-Contrast Reconstruction. *ISMRM 2021*.
    - Lin, A., Bilgic, B., & Ba, D. (2022). Bayesian Sensitivity Encoding Enables Parameter-Free, Highly Accelerated Joint Multi-Contrast Reconstruction. *ISMRM 2022*.
- Check out our code: https://github.com/al5250/sparse-bayes-learn