Semester Project

Compressive Sensing with Applications to Near-Field to Far-Field Antenna Pattern Extrapolation

Jonas Thalmeier

30/04/2025

EURECOM
Sophia Antipolis

Contents

1 Introduction

This semester project is carried out within the framework of the Data Science track of the Master program and is supervised by Prof. Dirk Slock, Dr. Zilu Zhao, and Dr. Fangqing Xiao. The project focuses on the theory and application of compressive sensing, a signal processing technique that allows for the recovery of sparse signals from undersampled measurements.

In the first phase (100-hour scope), the project explores state-of-the-art methods in compressive sensing, with particular attention to Sparse Bayesian Learning (SBL) and the use of Stein's Unbiased Risk Estimate (SURE) for adaptive hyperparameter tuning in the SBL prior. Theoretical understanding and simulation-based validation on synthetic data are key objectives of this part.

In the extended 200-hour version, the project further investigates the practical application of compressive sensing techniques to antenna pattern extrapolation, specifically in transitioning from near-field to far-field measurements. This includes the construction of appropriate dictionaries or sensing matrices, initially based on simplified analytical models of antenna radiation patterns.

2 Synthetic Data Generation

To evaluate the performance of the implemented algorithms in a controlled environment, synthetic data was generated following a standard compressive sensing model. The measurement vector $\mathbf{t} \in \mathbb{R}^N$ is constructed as

$$\mathbf{t} = \Phi\mathbf{w} + \mathbf{e}, \tag{1}$$

where $\Phi \in \mathbb{R}^{N \times D}$ is a sensing matrix, $\mathbf{w} \in \mathbb{R}^D$ is a sparse signal, and $\mathbf{e} \in \mathbb{R}^N$ is additive Gaussian noise with standard deviation σ.

The sensing matrix Φ is constructed by selecting N rows at random from a normalized $D \times D$ discrete Fourier transform (DFT) matrix. Optionally, a random Gaussian matrix with variance $1/N$ can be used instead of the DFT for comparison purposes.

The sparse vector $\mathbf{w}$ contains a user-defined fraction $\rho \in (0, 1)$ of non-zero entries. These entries are assigned random values drawn from a standard normal distribution, and their indices are randomly selected. The noise vector $\mathbf{e}$ is sampled from a Gaussian distribution $\mathcal{N}(0, \sigma^2)$.

This setup allows flexible control over the number of measurements N, signal dimensionality D, sparsity level ρ, and noise level σ, providing a reproducible and interpretable environment for algorithmic evaluation. The generation process is deterministic under a fixed random seed, ensuring consistency across multiple runs.

3 Choice of Algorithm

In the early phase of this project, a central task was to select a suitable inference algorithm for sparse signal recovery. The field of *Approximate Message Passing* (AMP) algorithms has seen considerable research activity in recent years, offering powerful and efficient solutions for compressive sensing problems. A concise yet comprehensive overview is provided in the tutorial by Zou and Yang [9], which summarizes key variants and their respective strengths.

However, AMP methods rely heavily on assumptions about the structure of the measurement matrix. As noted by Zou and Yang [9], AMP algorithms often fail to converge when the measurement matrix deviates from the independent and identically distributed (i.i.d.) sub-Gaussian regime. To address this limitation, modified approaches such as Orthogonal AMP (OAMP) have been proposed Zou and Yang [9], extending AMP's applicability to certain classes of unitarily-invariant matrices.

In the context of this project, the eventual goal is to apply compressive sensing to near-field to far-field antenna pattern extrapolation. The underlying sparse representation may involve FFT bases or modal

decompositions of the near-field, which are expected to yield structured, non-random measurement matrices. To maintain flexibility in handling such structure, methods constrained to i.i.d. random matrices—such as classical AMP—appear less suitable.

As a result, I chose to explore an alternative inference approach based on the Expectation Maximization (EM) algorithm, as introduced by Wipf and Rao [8] first. The EM-based SBL framework is known for its robustness and generality, making fewer assumptions about the measurement matrix. While it may not offer the same computational efficiency as AMP in ideal settings, it provides a more versatile and stable foundation for the anticipated applications of this project.

4   EM-Based Sparse Bayesian Learning

To estimate sparse weights from an overcomplete basis, the Expectation-Maximization (EM) algorithm proposed in Wipf and Rao [8] was implemented. This method belongs to the broader Sparse Bayesian Learning (SBL) framework, where sparsity is induced not by fixed priors, but through evidence maximization over a parameterized Gaussian prior.

4.1   Model Overview

Given an observation model
$$\mathbf{t} = \Phi\mathbf{w} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2\mathbf{I}), \tag{2}$$

where $\mathbf{t} \in \mathbb{R}^N$ is the observed signal, $\Phi \in \mathbb{R}^{N \times D}$ is the (possibly overcomplete) measurement matrix, and $\mathbf{w} \in \mathbb{R}^D$ is the weight vector to be estimated, SBL assumes a zero-mean Gaussian prior with individual variances for each weight:
$$p(\mathbf{w}; \boldsymbol{\gamma}) = \prod_{i=1}^{M} (2\pi\gamma_i)^{-\frac{1}{2}} \exp(-\frac{w_i^2}{2\gamma_i}). \tag{3}$$

The hyperparameters $\boldsymbol{\gamma}$ (the prior variances of the weights) are inferred via Type-II maximum likelihood by marginalizing out $\mathbf{w}$ and maximizing the resulting evidence.

4.2   Algorithm Description

The EM algorithm alternates between computing the posterior distribution over weights (E-step) and updating the hyperparameters (M-step). Specifically:

E-step: The posterior $p(\mathbf{w} \mid \mathbf{t}, \boldsymbol{\gamma}, \beta)$ is Gaussian:
$$\boldsymbol{\Sigma}_w = \left(\beta\Phi^\top\Phi + \mathrm{diag}(\boldsymbol{\gamma}^{-1})\right)^{-1}, \tag{4}$$
$$\boldsymbol{\mu}_w = \beta\boldsymbol{\Sigma}_w\Phi^\top\mathbf{t}, \tag{5}$$

where $\beta = 1/\sigma^2$ is the precision of the Gaussian likelihood.

M-step: The prior variances are updated based on the posterior moments:
$$\gamma_i = \mu_{w,i}^2 + (\Sigma_w)_{ii}, \tag{6}$$

and the noise variance is updated as:
$$\sigma^2 = \frac{1}{N}\|\mathbf{t} - \Phi\boldsymbol{\mu}_w\|^2 + \frac{1}{N}\sum_i (\Phi\Sigma_w\Phi^\top)_{ii}. \tag{7}$$

In my implementation, the update of σ^2 was simplified or deferred for stability, and a fixed β was used initially.

4.3 Implementation Remarks

The algorithm was implemented in Python based directly on the formulation in Wipf and Rao [8]. A simplified version of the EM loop is shown below:

```
mu, Sigma = estimate_posterior()
gamma = mu**2 + np.diag(Sigma)
```

The evolution of weights was tracked during training for diagnostic purposes, and convergence is monitored using the maximum change in μ and the relative mean squared error with respect to $\mathbf{t}$.

The main practical limitation of this approach lies in the need to invert a $D \times D$ matrix at every iteration to compute Σ_w, making it computationally expensive for high-dimensional problems. This is noted in the original paper as well [8, p. 2155], and future work may include reformulating the algorithm to reduce this complexity using the Woodbury identity or low-rank approximations.

4.4 Scalability via Covariance-Free Expectation Maximization (CoFEM)

While the EM-based Sparse Bayesian Learning implementation performs well for small-scale problems, it becomes prohibitively slow as the problem size increases. This is due to the need to invert a dense $D \times D$ matrix in each iteration to compute the posterior covariance. To address this computational bottleneck, the *Covariance-Free EM (CoFEM)* algorithm as proposed by Lin et al. [3] was implemented.

CoFEM avoids explicitly computing or storing the posterior covariance matrix. Instead, it estimates the required posterior statistics—the mean vector $\boldsymbol{\mu}_w$ and the diagonal elements of the covariance matrix Σ_{jj}—by solving a set of linear systems involving the precision matrix. These estimates are obtained using Rademacher probe vectors and linear solvers such as conjugate gradient, allowing for significant computational and memory savings.

In the E-step, the algorithm solves one linear system to compute the posterior mean and K additional systems to estimate the posterior variances using the diagonal estimation rule. The M-step then updates the hyperparameters α_j using the standard SBL update:

$$\alpha_j = \frac{1}{\mu_j^2 + s_j}, \tag{8}$$

where μ_j is the j-th component of the posterior mean and s_j is the estimated variance of w_j obtained from the Rademacher probe-based approximation.

Although the CoFEM algorithm is designed to benefit from conjugate gradient (CG) solvers for efficiently handling large-scale linear systems, the custom CG implementation used in this project proved to be slower than expected. In practice, it exhibited slower convergence and higher runtime compared to NumPy's built-in direct solver.. As a result, the final implementation resorts to NumPy's `np.linalg.solve` for solving the linear systems in the E-step, which yielded higher performance for the problem sizes considered in this study.

To assess the scalability, the runtime of the EM and CoFEM algorithm was measured as a function of the number of measurements N, with parameters set as follows: $D = 3N$, sparsity level $\rho = 0.1$, and noise variance $\sigma^2 = 10^{-6}$. The algorithms iterated until the relative residual dropped below 1e-2:

$$\frac{|\mathbf{t} - \Phi\boldsymbol{\mu}|_2}{|\mathbf{t}|_2} < 10^{-2} \tag{9}$$

Both algorithms required roughly the same numbers of iterations.

As shown in Figure 1, the CoFEM implementation significantly reduces runtime compared to the standard EM approach, especially as the dimensionality increases.
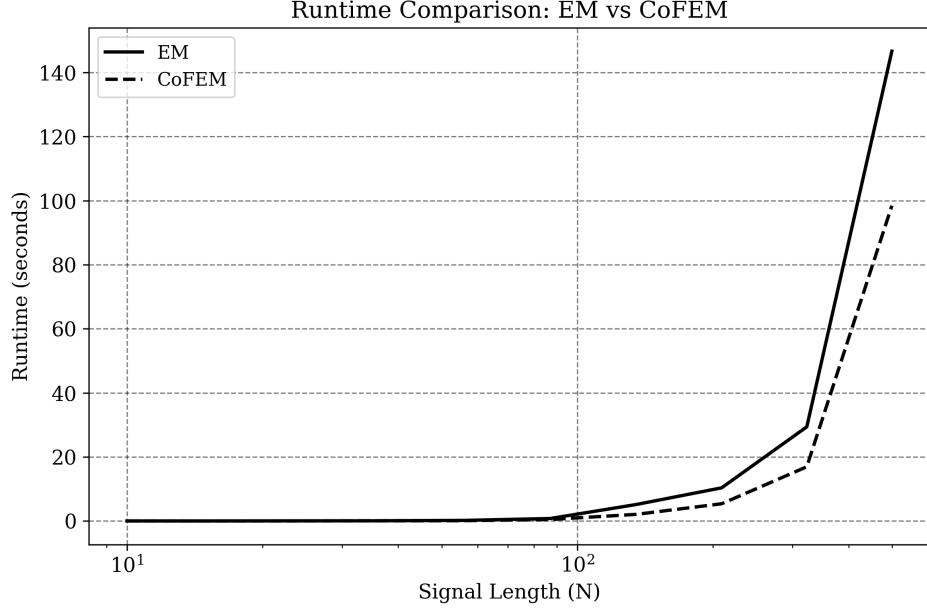
Figure 1: Runtime of the EM and CoFEM algorithms for increasing number of measurements N. The dictionary size is set to $D = 3N$.

While the CoFEM algorithm achieves a noticeable speedup over the standard EM algorithm—98.4 seconds compared to 146.6 seconds for N=500—the improvement falls short of the expected at least twofold acceleration.

4.5   Performance Comparison of EM Algorithms

Gaussian Matrix   To compare the performance of different EM-based Sparse Bayesian Learning algorithms, experiments were conducted using synthetic data generated from Gaussian dictionaries, varying both the sparsity level ρ and the undersampling factor $\delta = N/D$. Four variants were considered: standard EM and CoFEM, each with known and unknown noise variance. The normalized root mean squared error (NRMSE) was used to evaluate reconstruction quality. Results are averaged over multiple random trials to ensure statistical robustness.

The signal dimension was fixed to $D = 1024$. For the sparsity analysis the undersampling factor was fixed at $\delta = 0.25$, corresponding to $N = \lfloor \delta D \rfloor$ measurements. For the undersampling analysis, the sparsity level was fixed at $\rho = 0.06$. The measurement matrix was drawn from a standard Gaussian distribution. Additive Gaussian noise with standard deviation $\sigma = 0.01$ was added to the measurements. For each setting, the reported results represent the average over five random trials to account for variability due to random initialization and noise.

As shown in Figure 2 and Figure 3, all four algorithms exhibit remarkably similar performance across the full range of tested sparsity and undersampling settings. In particular, the CoFEM algorithm with unknown noise variance performs on par with the variants that assume knowledge of the noise level. This observation is consistent with the findings reported in Lin et al. [3], where CoFEM and EM methods were shown to yield nearly identical reconstruction accuracy under equivalent settings.
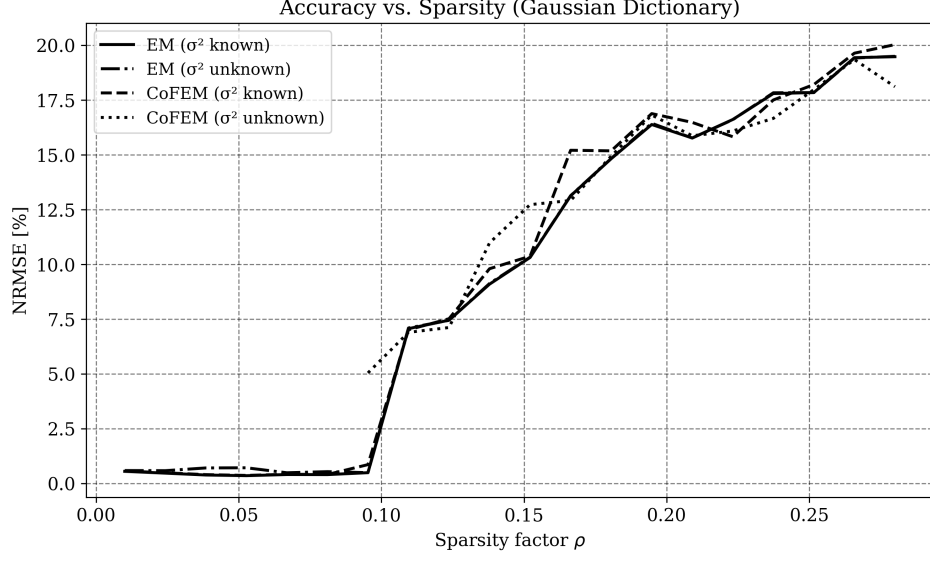
Figure 2: Reconstruction accuracy (NRMSE) versus sparsity level ρ. All four EM variants show near-identical behavior.
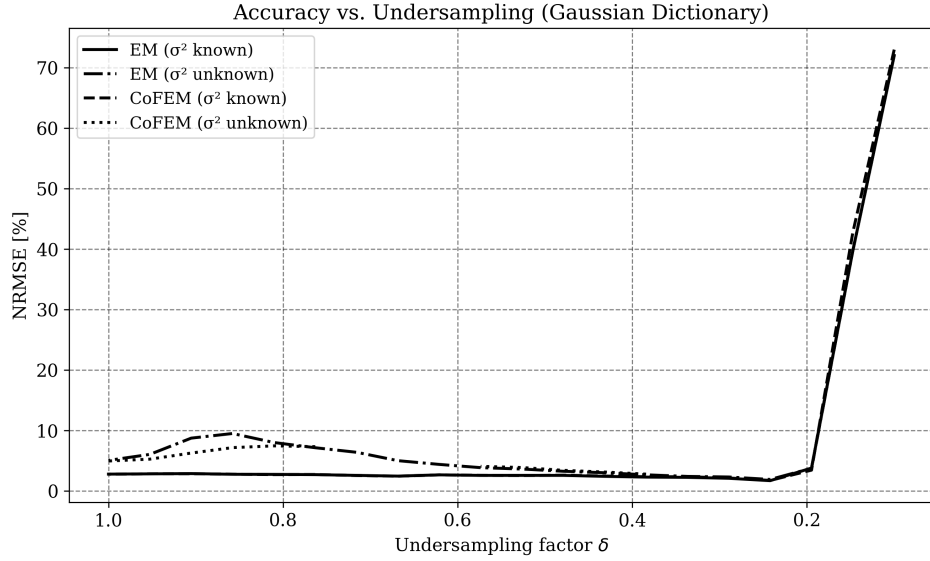

Figure 3: Reconstruction accuracy (NRMSE) versus undersampling factor δ. Similar performance is observed across all algorithms.

Given these results, and in order to simplify subsequent analysis, I choose to focus on the CoFEM algorithm with unknown noise variance in the remainder of this report. This choice is motivated by its computational scalability, the practical advantage of not requiring prior knowledge of the noise level, and the observation that its reconstruction performance closely matches that of other EM-based approaches in the tested settings. While this similarity may not generalize to all scenarios, it provides a reasonable basis for concentrating on a single representative method in this context.

Fourier Matrix   When the measurement matrix Φ is changed from a Gaussian random matrix to one representing a subsampled Fourier transform, several complications arise. Both the EM and CoFEM algorithms

5

were adapted accordingly, replacing standard transposes with conjugate transposes and applying further minor modifications to support complex-valued inputs. Multiple approaches were explored to enable accurate diagonal estimation within the CoFEM framework under complex-valued settings; however, none of them yielded satisfactory results.

Despite these efforts, the EM algorithm continued to show unstable behavior. The initialization of the hyperparameter vector $\boldsymbol{\alpha}$ was found to significantly influence the outcome. In some instances, the algorithm converged to solutions with accuracy comparable to those obtained in the Gaussian matrix setting (Figure 5). In other cases, however, it resulted in drastically increased MSE values (Figure 4). This instability, combined with the sensitivity to initialization, rendered the EM approach unreliable for complex-valued measurement matrices.

In contrast, the algorithm described in Section 5, which incorporates Sparse Bayesian Learning with SURE-based hyperparameter optimization, demonstrated more stable and consistent performance in the complex-valued setting. As a result, further efforts to adapt the EM algorithm for this case were discontinued.
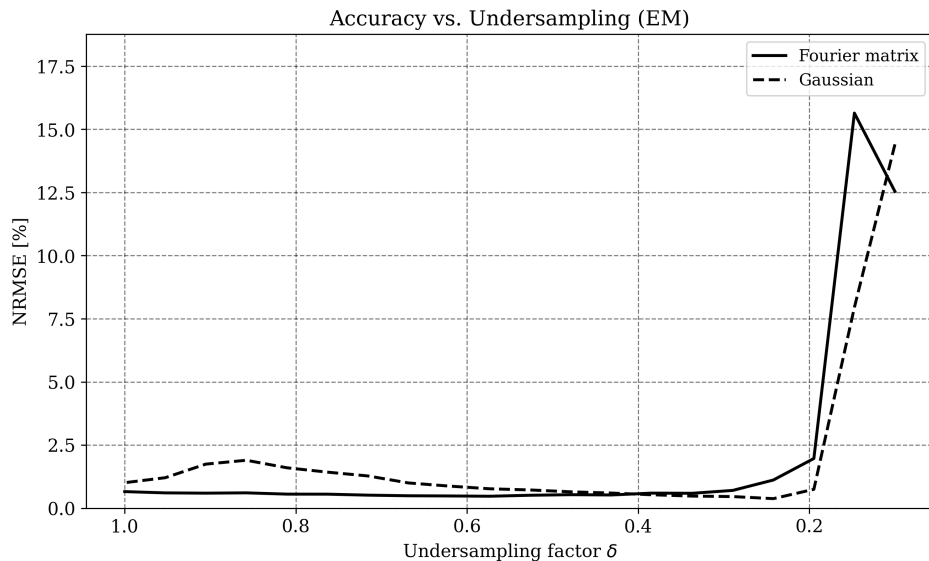

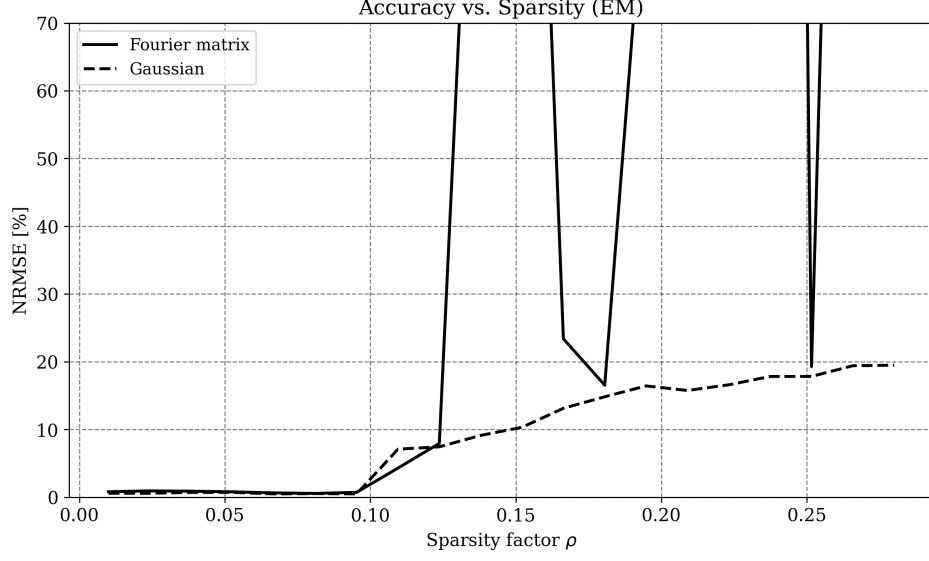Figure 4: Reconstruction accuracy (NRMSE) versus undersampling factor δ.

Figure 5: Reconstruction accuracy (NRMSE) versus undersampling factor δ.

5 Sparse Baysian Learning with Stein's Unbiased Risk Estimate

5.1 Stein's Lemma

Stein's Lemma provides a fundamental relationship between the expectation of a function of a Gaussian random variable and its derivative. For the univariate case, let $Z \sim \mathcal{N}(0,1)$ and $f : \mathbb{R} \to \mathbb{R}$ be an absolutely continuous function with derivative f' satisfying $\mathbb{E}|f'(Z)| < \infty$. Then, Stein's Lemma states:

$$\mathbb{E}[Zf(Z)] = \mathbb{E}[f'(Z)] \tag{10}$$

The proof follows from integration by parts and leverages the property of the standard normal density $\phi(z)$, where $\phi'(z) = -z\phi(z)$ (Tibshirani and Wasserman [6]). This result extends to $X \sim \mathcal{N}(\mu, \sigma^2)$ through standardization:

$$\frac{1}{\sigma^2}\mathbb{E}[(X - \mu)f(X)] = \mathbb{E}[f'(X)] \tag{11}$$

For the multivariate case, let $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2\mathbf{I}_n)$ and let $f : \mathbb{R}^n \to \mathbb{R}$ be an almost differentiable function with gradient ∇f. Stein's Multivariate Lemma states:

$$\frac{1}{\sigma^2}\mathbb{E}[(\mathbf{X} - \boldsymbol{\mu})f(\mathbf{X})] = \mathbb{E}[\nabla f(\mathbf{X})] \tag{12}$$

This is proved by applying the univariate lemma component-wise and using Fubini's theorem under the almost differentiability condition Tibshirani and Wasserman [6]. The lemma enables covariance estimation without explicit knowledge of $\boldsymbol{\mu}$, replacing it with computable partial derivatives.

5.2 Stein's Unbiased Risk Estimate (SURE)

For a Gaussian model $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2\mathbf{I}_n)$ and estimator $\hat{\boldsymbol{\mu}}(\mathbf{y})$, the risk $R = \mathbb{E}\|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\|_2^2$ can be decomposed as:

$$R = -n\sigma^2 + \mathbb{E}\|\mathbf{y} - \hat{\boldsymbol{\mu}}\|_2^2 + 2\sigma^2 \sum_{i=1}^{n} \text{Cov}(y_i, \hat{\mu}_i) \tag{13}$$

Using Stein's Lemma, the covariance term is replaced with partial derivatives when $\hat{\boldsymbol{\mu}}$ is almost differentiable. This yields Stein's Unbiased Risk Estimate (SURE):

$$\widehat{R} = -n\sigma^2 + \|\mathbf{y} - \hat{\boldsymbol{\mu}}\|_2^2 + 2\sigma^2 \sum_{i=1}^{n} \frac{\partial \hat{\mu}_i}{\partial y_i}(\mathbf{y}) \tag{14}$$

SURE provides an unbiased risk estimator ($\mathbb{E}[\widehat{R}] = R$) without requiring knowledge of $\boldsymbol{\mu}$ ([6]). For estimators depending on tuning parameters λ, SURE enables model selection through:

$$\hat{\lambda} = \underset{\lambda \in \Lambda}{\arg\min} \left(\|\mathbf{y} - \hat{\boldsymbol{\mu}}_\lambda\|_2^2 + 2\sigma^2 \sum_{i=1}^{n} \frac{\partial \hat{\mu}_{\lambda,i}}{\partial y_i}(\mathbf{y}) \right) \tag{15}$$

This framework has become instrumental for risk estimation and hyperparameter optimization in high-dimensional statistics.

5.3 Algorithm Design for SBL with SURE

The implementation of Sparse Bayesian Learning (SBL) with Stein's Unbiased Risk Estimator (SURE) was planned based on the theoretical framework presented in Slock [5]. In this setting, we consider the linear model:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{v}, \quad \mathbf{v} \sim \mathcal{N}(\mathbf{v}; \mathbf{0}, \sigma^2 \mathbf{I}), \quad \mathbf{x} \sim \mathcal{N}(\mathbf{x}; \mathbf{0}, \mathbf{P}), \tag{16}$$

where $\mathbf{y}$ represents the observed data, $\mathbf{A}$ is the measurement matrix, $\mathbf{x}$ is the sparse signal of interest, and $\mathbf{v}$ is additive Gaussian noise. The prior on $\mathbf{x}$ is Gaussian with covariance $\mathbf{P}$, which is typically diagonal to enforce sparsity. The posterior distribution of $\mathbf{x}$ given $\mathbf{y}$ is derived as:

$$\mathbf{x}|\mathbf{y} \sim \mathcal{N}\left(\mathbf{x}; \mathbf{P}\mathbf{A}^T \mathbf{R}^{-1}\mathbf{y}, \mathbf{P} - \mathbf{P}\mathbf{A}^T \mathbf{R}^{-1}\mathbf{A}\mathbf{P}\right), \tag{17}$$

where $\mathbf{R} = \mathbf{A}\mathbf{P}\mathbf{A}^T + \sigma^2 \mathbf{I}$. Alternatively, this can be rewritten as:

$$\mathbf{x}|\mathbf{y} \sim \mathcal{N}\left(\mathbf{x}; \mathbf{S}^{-1}\mathbf{A}^T \mathbf{y}, \sigma^2 \mathbf{S}^{-1}\right), \tag{18}$$

with $\mathbf{S} = \mathbf{A}^T \mathbf{A} + \sigma^2 \mathbf{P}^{-1}$.

5.4 Connection to Tipping's Fast Marginal Likelihood Algorithm

The posterior expressions derived above align with the sparse Bayesian framework described in Tipping and Faul [7]. By adopting the notation used in Tipping's paper, where $\mathbf{A} \to \boldsymbol{\Phi}$, $\mathbf{y} \to \mathbf{t}$, $\mathbf{x} \to \mathbf{w}$, and $\mathbf{P} \to \mathbf{A}^{-1}$, the posterior mean and covariance become:

$$\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Phi}^T \mathbf{t}, \quad \boldsymbol{\Sigma} = \left(\mathbf{A} + \sigma^{-2}\boldsymbol{\Phi}^T \boldsymbol{\Phi}\right)^{-1}. \tag{19}$$

These expressions are identical to those derived in Tipping and Faul [7] for the posterior distribution of the weights $\mathbf{w}$ in the sparse Bayesian learning framework. This equivalence confirms that the Fast Marginal

Likelihood Maximisation algorithm proposed by Tipping can be directly applied to implement SBL with SURE in our setting.

The algorithm proceeds as follows:

1. Initialize the hyperparameters $\boldsymbol{\alpha}$ and noise variance σ^2.

2. Initialize the set of active basis $\mathcal{A}$ and compute the posterior statistics $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ using the current hyperparameters.

3. Evaluate the marginal likelihood and update $\boldsymbol{\alpha}$ by sequentially adding, deleting, or re-estimating basis functions to maximize the marginal likelihood.

4. Iterate until convergence, ensuring that the hyperparameters and noise variance are optimized for minimal mean squared error (MSE) as guided by SURE.

This approach leverages the efficiency of Tipping's algorithm while incorporating the risk estimation principles of SURE to optimize hyperparameters. The result is a computationally tractable implementation of SBL that maintains sparsity and generalizes well, as demonstrated in the experiments reported in Tipping and Faul [7].

5.5 Implementation of the Fast Marginal Likelihood Algorithm

The Fast Marginal Likelihood Algorithm is implemented through an efficient basis selection and pruning strategy. Unlike traditional SBL approaches that update all basis functions simultaneously, this implementation sequentially evaluates individual basis functions to maximize the marginal likelihood more efficiently.

The algorithm begins by initializing a single most relevant basis function, selected by evaluating the correlation between each basis vector and the target signal. All other basis functions are initially deactivated by setting their corresponding α values to infinity. This sparse initialization provides a computationally efficient starting point.

$$
\begin{aligned}
\boldsymbol{\alpha_i} &= \infty \\
\sigma^2 &= 1/\beta = 0.1\sigma_{\boldsymbol{t}}^2 \\
\theta_i &= \frac{\|\phi_i^* t\|}{\Re\{\phi_i^* \phi_i\}} \\
\mathcal{A} &= \arg\max_i \theta_i \\
\alpha_{i \in \mathcal{A}} &= \frac{\Re\{\phi_i^* \phi_i\}}{\theta_i - \sigma^2} \\
\Sigma &= (diag\{\alpha_{\mathcal{A}}\} + \beta\Phi_{\mathcal{A}}^H \Phi_{\mathcal{A}})^{-1} \\
\mu &= \beta\Sigma\Phi_{\mathcal{A}}^H \boldsymbol{t}
\end{aligned}
\tag{20}
$$

Where ϕ_j is the j-th column of Φ and $\alpha_{\mathcal{A}}/\Phi_{\mathcal{A}}$ the hyperparameters/basis that are contained in the active basis $\mathcal{A}$

During each iteration, the algorithm considers three possible actions for each basis function:

- Adding a currently inactive basis (if $\alpha = \infty$ and $q_i^2 < s_i$)

- Deleting an active basis from the model (if $\alpha_i < \infty$ and $q_i^2 < s_i$)

- Re-estimating the α value of an active basis (if $\alpha > \infty$ and $q_i^2 > s_i$)

For each potential action, the algorithm computes the change in marginal likelihood (ΔL) that would result. This computation involves two key quantities for each basis i:

- The sparsity factor s_i, measuring the potential sparsity contribution

$$
s_i = \begin{cases} \frac{\alpha_i S_i}{\alpha_i - S_i} & , \alpha_i < \infty \\ S_i & , \alpha_i = \infty \end{cases}
\tag{21}
$$

9

- The quality factor q_i, indicating the basis function's alignment with the target

$$q_i = \begin{cases} \frac{\alpha_i Q_i}{\alpha_i - S_i} & , \alpha_i < \infty \\ Q_i & , \alpha_i = \infty \end{cases} \tag{22}$$

where:

$$\begin{aligned} S_i &= \beta \|\phi_i\|^2 - \beta^2 \|\Sigma(\phi_{\mathcal{A}} \Phi)\|^2 \\ Q_i &= \beta \phi_i^H \boldsymbol{t} - \beta^2 \phi_i^H \Phi_{\mathcal{A}} \Sigma \Phi_{\mathcal{A}}^H \boldsymbol{t} \end{aligned} \tag{23}$$

These factors are efficiently computed using the current model state without requiring full matrix inversions. The algorithm then selects the action that yields the largest positive change in marginal likelihood If no action provides sufficient improvement ($\Delta L <$ threshold), the algorithm terminates.

$$\begin{aligned} \Delta \mathcal{L}_i^{add} &= \frac{1}{2} \left(\frac{Q_i^2 - S_i}{S - i} + \log \frac{S_i}{Q_i^2} \right) \\ \Delta \mathcal{L}_i^{re-estimate} &= \frac{1}{2} \left(\frac{Q_i^2}{S_i + [\tilde{\alpha}_i^{-1} - \alpha_i^{-1}]^{-1}} - \log(1 + S_i[\tilde{\alpha}_i^{-1} - \alpha_i^{-1}]) \right) \\ \Delta \mathcal{L}_i^{delete} &= \frac{1}{2} \left(\frac{Q_i^{-1}}{S_i - \alpha_i} - \log \left(1 - \frac{S_i}{\alpha_i} \right) \right) \end{aligned} \tag{24}$$

The posterior statistics ($\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$) are maintained and updated after each action. For addition and re-estimation, only the affected elements of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ need to be recomputed. When deleting a basis, the algorithm efficiently downdates the model parameters.

This sequential approach provides several advantages:

- Computational efficiency through selective updates

- Natural sparsity promotion through explicit basis selection

- Robust convergence through guaranteed marginal likelihood improvements

- Automatic determination of model complexity

The implementation handles both real and complex-valued inputs, with appropriate conjugate operations for complex matrices. The noise precision β ($1/\sigma^2$) can either be specified or estimated from the data during optimization.

This efficient implementation enables the algorithm to handle high-dimensional problems while maintaining sparsity and achieving fast convergence, typically requiring fewer iterations than traditional SBL approaches.

6   Comparative Analysis of EM and SBL-SURE Methodologies

6.1   Accuracy vs. Undersampling and Sparsity

To evaluate the performance of both the SBL-SURE framework and the EM algorithm, we replicated the experimental conditions outlined in Section 4.5. While EM and CoFEM exhibited nearly identical reconstruction accuracy in Gaussian measurement systems, the EM algorithm proved unstable for complex-valued (FFT-based) dictionaries, frequently failing to converge. Consequently, EM results for the complex case were excluded from this comparison. Notably, in the rare instances where EM did converge for complex systems, its reconstruction error aligned with its Gaussian performance, suggesting no inherent disadvantage in complex domains—only algorithmic instability.

Tipping's Fast Marginal Likelihood algorithm, however, demonstrated robust performance across both Gaussian and complex-valued systems. As shown in Figures 6 and 7, its accuracy in Gaussian systems closely

matched EM's, with EM achieving marginal improvements (2–4% NRMSE reduction or $\Delta\rho \approx 0.05$). Strikingly, in complex-valued systems, Tipping's method outperformed its own Gaussian results—a paradoxical reversal of trends observed with EM. This suggests that the Fast Marginal Likelihood framework inherently adapts more effectively to structured dictionaries like Fourier bases, despite EM's theoretical equivalence.
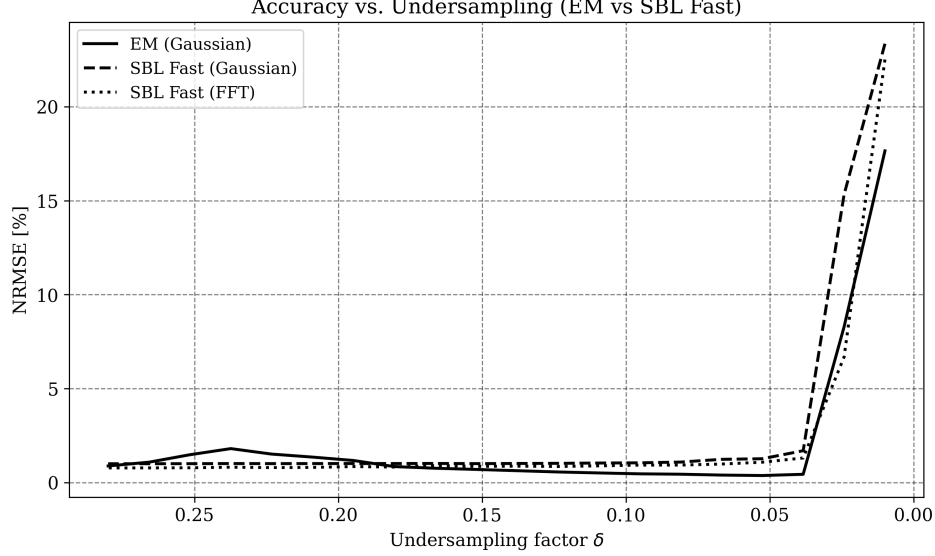

Figure 6: Reconstruction accuracy (NRMSE) versus undersampling factor δ. Tipping's algorithm maintains consistent performance across Gaussian and complex systems, while EM (excluded here for instability) was restricted to Gaussian cases.
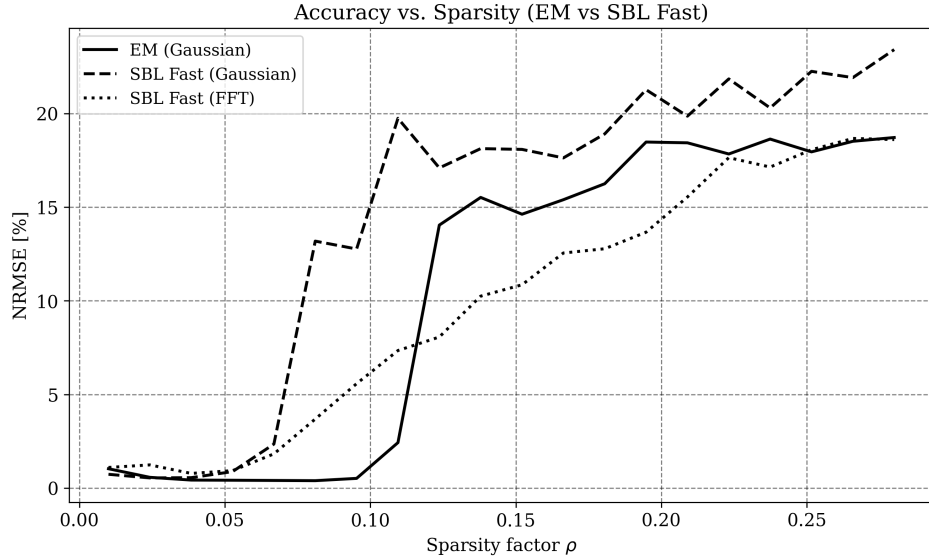

Figure 7: Reconstruction accuracy (NRMSE) versus ground-truth sparsity ratio ρ. Tipping's method excels in complex systems, achieving lower NRMSE than in Gaussian cases—a reversal of EM's behavior.

6.2 Computational Efficiency

Figure 8 compares the execution times of EM, CoFEM, and the Fast Marginal Likelihood algorithm under standardized conditions:

- Sparsity ratio $\rho = 0.1$

- Noise variance $\sigma^2 = 10^{-6}$

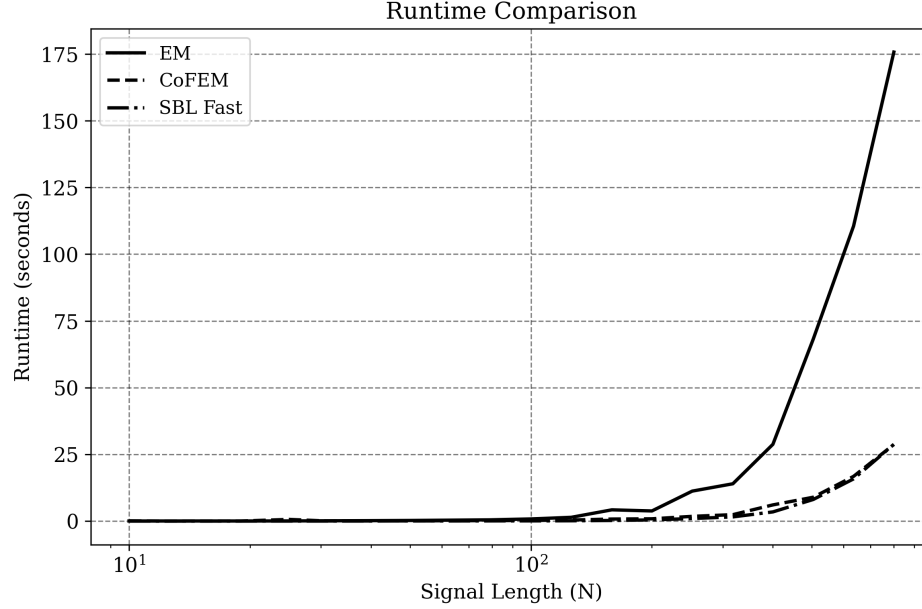- Convergence criterion $\frac{|\mathbf{t} - \Phi\boldsymbol{\mu}|_2}{|\mathbf{t}|_2} < 10^{-2}$

Figure 8: Runtime versus measurement count N for Gaussian dictionaries ($D = 3N$). The Fast Marginal Likelihood algorithm exhibits the same computational efficiency as CoFEM and achieves significant speedups over EM, particularly at larger scales.

6.3 Sparsity

A critical distinction lies in how each algorithm enforces sparsity. Tipping's method explicitly adds or prunes basis functions, yielding strictly sparse solutions. In contrast, EM gradually attenuates irrelevant weights toward near-zero values without exact truncation. Under low-noise conditions ($\sigma = 0.01$), both methods successfully identify the true support while suppressing extraneous coefficients, as shown in Figures 9 and 10.
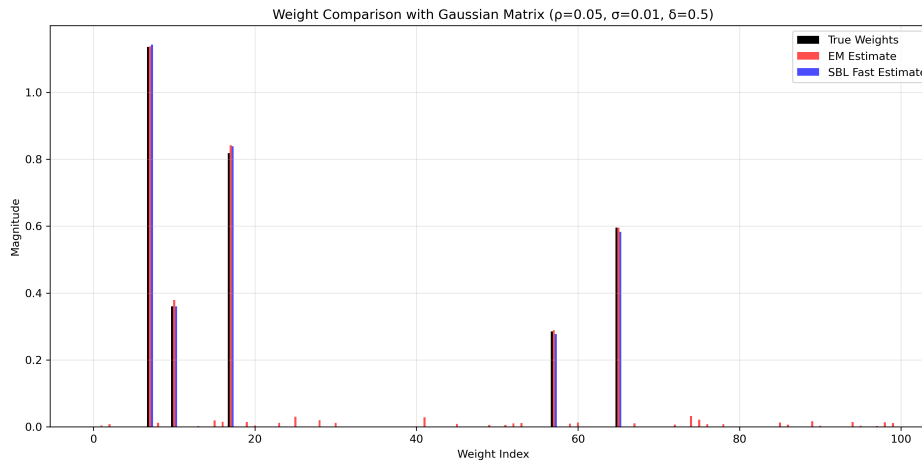

Figure 9: Weight magnitude comparisons (Gaussian dictionary, $\sigma = 0.01$). Both algorithms recover the true support, though SBL produces strictly sparse estimates.
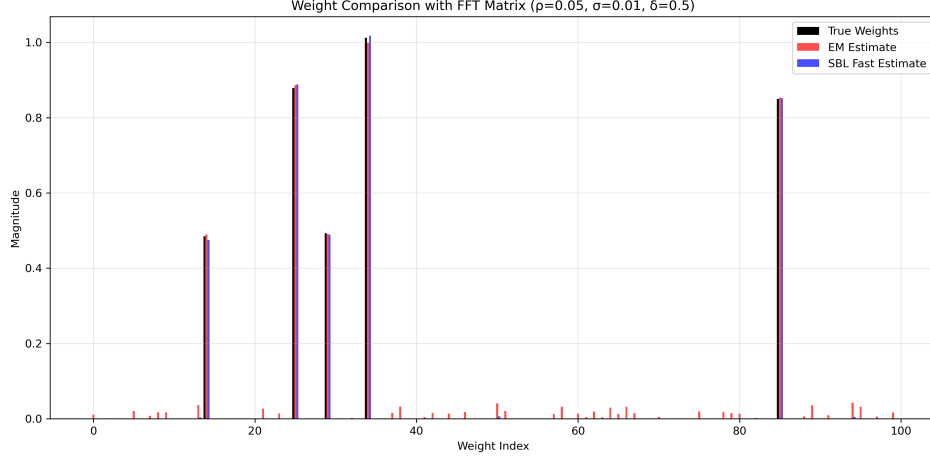
12

Figure 10: Weight magnitude comparisons (complex-valued dictionary, $\sigma = 0.01$). SBL's explicit basis selection enhances sparsity enforcement in Fourier-like systems.

At higher noise levels ($\sigma = 0.01$), Figures 11 and 12 reveal comparable challenges: neither algorithm reliably identifies the true support, though SBL maintains stricter sparsity through its deletion mechanism.
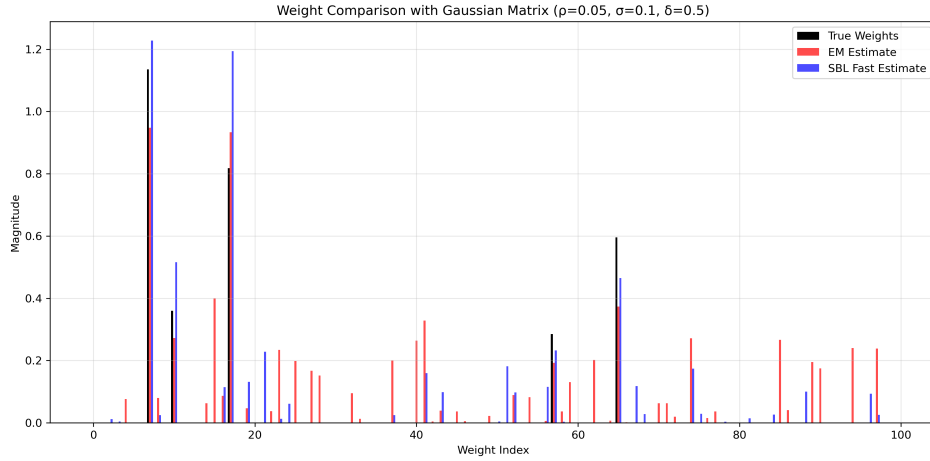

Figure 11: Weight estimates under moderate noise (Gaussian dictionary, $\sigma = 0.01$). Both methods struggle with support recovery, though SBL avoids non-zero artifacts more effectively.
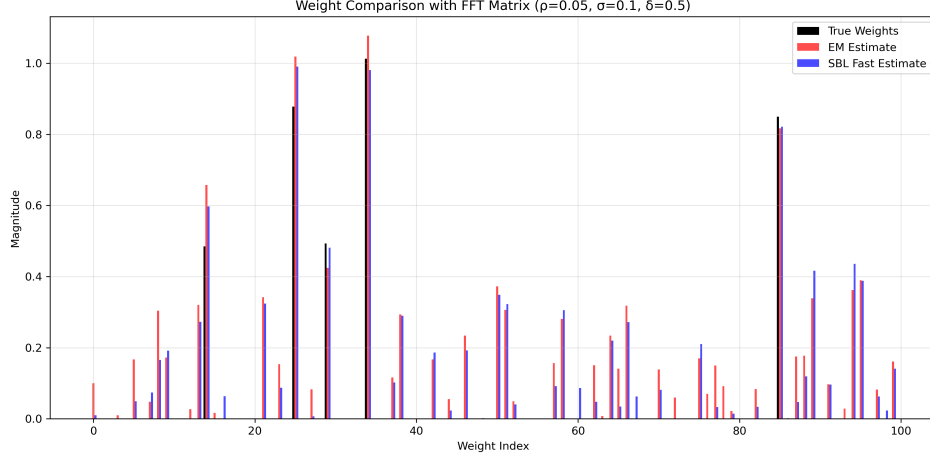
13

Figure 12: Weight estimates under moderate noise (complex-valued dictionary, $\sigma = 0.01$). Both methods struggle with support recovery, though SBL avoids non-zero artifacts more effectively.

7 Spherical Near-Field Antenna Patterns

7.1 Spherical-Wave Expansion

Spherical wave expansion (SWE) provides a mathematical framework for analyzing electromagnetic fields, particularly those radiated by antennas. By decomposing the electric field $\mathbf{E}(r, \theta, \phi)$ into a sum of orthogonal spherical wave functions $\mathbf{m}_{e,o_{mn}}$ and $\mathbf{n}_{e,o_{mn}}$, SWE enables a systematic representation of the field in terms of outward-traveling waves. The expansion coefficients $a_{e,o_{mn}}$ and $b_{e,o_{mn}}$ are determined by projecting the tangential components of the field onto these basis functions, as shown in the integral expressions. This approach is especially useful for antenna characterization, near-field to far-field transformations, and scattering analysis, as it captures the field's angular and radial dependencies in a compact and physically meaningful way. The use of Hankel functions of the second kind ensures that only outward-propagating waves are considered, making SWE well-suited for problems involving radiation from finite sources. Ludwig [4] presents how to derive the coefficients $a_{e,o_{mn}}$ (TE waves) and $b_{e,o_{mn}}$ (TM waves) in order to present the electric field like:

$$\mathbf{E}(r, \theta, \phi) = -\sum_{m=-n}^{n} \sum_{n=1}^{\infty} a_{e,o_{mn}} \mathbf{m}_{e,o_{mn}} + b_{e,o_{mn}} \mathbf{n}_{e,o_{mn}} \tag{25}$$

where the harmonic functions are defined as

$$
\begin{aligned}
\mathbf{m}_{e,o_{mn}} = & \mp z_n(kr) \frac{m P_n^m(\cos\theta)}{\sin\theta} \frac{\sin}{\cos} m\phi\, \hat{\boldsymbol{\theta}} \\
& - z_n(kr) \frac{\partial}{\partial\theta} P_n^m(\cos\theta) \frac{\cos}{\sin} m\phi \hat{\boldsymbol{\phi}} \\
\mathbf{n}_{e,o_{mn}} = & n(n+1) \frac{z_n(kr)}{kr} P_n^m(\cos\theta) \frac{\sin}{\cos} m\phi \hat{\boldsymbol{r}} \\
& + \frac{1}{kr} \frac{\partial}{\partial r} \left[r z_n(kr) \right] \frac{\partial}{\partial\theta} P_n^m(\cos\theta) \frac{\sin}{\cos} m\phi \hat{\boldsymbol{\theta}} \\
& \pm \frac{1}{kr} \frac{\partial}{\partial r} \left[r z_n(kr) \right] \frac{m P_n^m(\cos\theta)}{\sin\theta} \frac{\cos}{\sin} m\phi \hat{\boldsymbol{\phi}}
\end{aligned}
\tag{26}
$$

where $P_n^{|m|}(\cos\theta)$ is the associated Legendre function. The radial function $z_n^{(c)}(kr)$ depend on the wave type c and are defined as:

$$
z_n^{(c)}(kr) = \begin{cases} j_n(kr) & , c = 1 \\ n_n(kr) & , c = 2 \\ h_n^{(1)}(kr) = j_n(kr) + in_n(kr) & , c = 3 \\ h_n^{(2)}(kr) = j_n(kr) - in_n(kr) & , c = 4 \end{cases}
\tag{27}
$$

In this case the Hankel function of second kind $(j_n - iy_n)$ is used as z_n, since only outward travelling waves are considered and the sources are assumed to have finite extend. The coefficients $a_{e,o_{mn}}$ and $b_{e,o_{mn}}$ an be obtained by solving the following integrals:

$$
\begin{aligned}
a_{e,0_{mn}} &= \frac{1}{[z_n(kr_1)]^2} \frac{2n+1}{\pi 2n(n+1)} \frac{(n-m)!}{(n+m)!} \\
&\quad \cdot \int_0^{2\pi} \int_0^\pi -\mathbf{m}_{e,o_{mn}} \cdot \mathbf{E}(r_1,\theta,\phi)_{\tan} \sin\theta \, d\theta \, d\phi \\
b_{e,o_{mn}} &= \frac{1}{[(1/kr_1)(\partial/\partial r)[r_1 z_n(kr_1)]]^2} \frac{2n+1}{\pi 2n(n+1)} \frac{(n-m)!}{(n+m)!} \\
&\quad \cdot \int_0^{2\pi} \int_0^\pi -\mathbf{n}_{e,o_{mn}} \cdot \mathbf{E}(r_1,\theta,\phi)_{\tan} \sin\theta \, d\theta \, d\phi
\end{aligned}
\tag{28}
$$

Initial trials applied the formulas from Ludwig [4] to expand the field of a simulated dipole antenna into spherical wave coefficients $a_{e,o_{mn}}$ and $b_{e,o_{mn}}$. The reconstructed field, derived from these coefficients, exhibited identical spatial patterns to the original field in 3D visualizations, but the magnitude scaling was inconsistent. Despite extensive literature review, the discrepancy remained unresolved. These coefficients would have served as a critical reference for validating subsequent Sparse Bayesian Learning results.

7.2 SBL for Near Field Patterns

Hofmann et al. [2] showed that the near-field of an antenna can be characterized with a minimum number of measurements under the assumption of sparsity in the spherical harmonics domain. The approach is based on expressing the near-field $\mathbf{E}(r,\theta,\phi)$ as a linear combination of spherical vector wave functions $\mathbf{h}_{smn}^{(4)}$. The theoretical framework is described in detail in Hansen [1]. In the following sections, the notation of Hansen will be used. The near-field is expressed as:

In this notation, $s = 1$ corresponds to $\mathbf{n}$-vector functions and $s = 2$ to the $\mathbf{m}$-vector functions. The use of even e and odd o modes is omitted by replacing sin and cos with complex exponential functions, reducing four real functions to two complex ones. The normalization is also modified. However, this notation is less common, and no explicit formulas for expansion via integration were found. Attempts to derive them independently were unsuccessful, resulting in incorrect magnitudes in the reconstructed field. Despite this, the compactness of the notation made it preferable for Sparse Bayesian Learning (SBL).

$$
\mathbf{E}(r,\theta,\phi) = \frac{k}{\sqrt{\eta}} \sum_{c=3}^4 \sum_{s=1}^2 \sum_{n=1}^\infty \sum_{m=-n}^m Q_{smn}^{(c)} \mathbf{F}_{smn}^{(c)}(r,\theta,\phi).
\tag{29}
$$

$\mathbf{E}(r,\theta,\phi)$ will correspond to $\mathbf{t}$ in the SBL framework. Analogously, the spherical vector wave functions $\mathbf{F}_{smn}^{(c)}(r,\theta,\phi)$ will be denoted as Φ and $Q_{smn}^{(c)}$ as $\mathbf{w}$.

$Q_{smn}^{(c)}$ are called the wave coefficients and are the unknowns to be estimated. η is the specific admittance and defined as $\sqrt{\epsilon/\mu}$. $c = 1$ and $c = 2$ indicate standing waves, while $c = 3$ represents an outward travelling wave, and $c = 4$ an inward travelling wave. Since we regard only antennas in transmission mode in the abscence of refelction, only $c = 4$ will be regarded. $s = 1$ denotes the m-function while $s = 2$ gives the n-function. The

15

spherical vector wave functions $\mathbf{F}^{(c)}_{smn}(r,\theta,\phi)$ are defined as:

$$\mathbf{F}^{(c)}_{1mn}(r,\theta,\phi) = \frac{1}{\sqrt{2\pi}}\frac{1}{\sqrt{n(n+1)}}\left(-\frac{m}{|m|}\right)^m$$

$$\left\{ z_n^{(c)}(kr)\frac{im\overline{P}_n^{|m|}(\cos\theta)}{\sin\theta}e^{im\phi}\hat{\boldsymbol{\theta}} \right.$$

$$\left. - z_n^{(c)}(kr)\frac{\mathrm{d}\overline{P}_n^{|m|}(\cos\theta)}{\mathrm{d}\theta}e^{im\phi}\hat{\boldsymbol{\phi}} \right\}$$

$$\mathbf{F}^{(c)}_{2mn}(r,\theta,\phi) = \frac{1}{\sqrt{2\pi}}\frac{1}{\sqrt{n(n+1)}}\left(-\frac{m}{|m|}\right)^m \tag{30}$$

$$\left\{ \frac{n(n+1)}{kr}z_n^{(c)}(kr)\bar{P}_n^{|m|}(\cos\theta)e^{im\phi}\hat{\boldsymbol{r}} \right.$$

$$+ \frac{1}{kr}\frac{d}{d(kr)}\left(krz_n^{(c)}(kr)\right)\frac{d\bar{P}_n^{|m|}(\cos\theta)}{d\theta}e^{im\phi}\hat{\boldsymbol{\theta}}$$

$$\left. + \frac{1}{kr}\frac{d}{d(kr)}\left(krz_n^{(c)}(kr)\right)\frac{im\bar{P}_n^{|m|}(\cos\theta)}{\sin\theta}e^{im\phi}\hat{\boldsymbol{\phi}} \right\}$$

where $\bar{P}_n^{|m|}(\cos\theta)$ is the normalized associated Legendre function:

$$\bar{P}_n^m(\cos\theta) = \sqrt{\frac{2n+1}{2}\frac{(n-m)!}{(n+m)!}}P_n^m(\cos\theta) \tag{31}$$

The derivation of the normalized associated Legendre function can be expressed as:

$$\frac{d\bar{P}_n^{|m|}(\cos\theta)}{d\theta} = \frac{d\bar{P}_n^{|m|}(\cos\theta)}{d\cos\theta}\frac{d\cos\theta}{d\theta}$$

$$= -\sin\theta\sqrt{\frac{2n+1}{2}\frac{(n-m)!}{(n+m)!}}\frac{dP_n^{|m|}(\cos\theta)}{d\cos\theta} \tag{32}$$

The radial function $z_n^{(c)}(kr)$ depend on the wave type c and are defined as:

$$z_n^{(c)}(kr) = \begin{cases} j_n(kr) & ,c=1 \\ n_n(kr) & ,c=2 \\ h_n^{(1)}(kr) = j_n(kr) + in_n(kr) & ,c=3 \\ h_n^{(2)}(kr) = j_n(kr) - in_n(kr) & ,c=4 \end{cases} \tag{33}$$

7.3 Implementation

In the SBL framework, the spherical harmonics domain is embedded by representing the measured electromagnetic field as a linear combination of spherical vector wave functions, which serve as the basis functions. The measurement matrix Φ (or F in the code) has dimensions $(N, D, 3)$, where N is the number of spatial sampling points (each corresponding to a unique pair of angles (θ,ϕ)), D is the total number of spherical harmonics modes (indexed by (n, m, s)), and the third dimension corresponds to the three Spherical field components (e.g., E_r, E_θ, E_ϕ). The measurement vector $\mathbf{t}$ is initially a matrix of shape $(N, 3)$, containing the three field components at each spatial point. To apply the SBL algorithm, which requires a standard linear model with scalar-valued entries, both Φ and $\mathbf{t}$ are reshaped: Φ is transformed to a matrix of shape $(3N, D)$ by stacking the field components for all points, and $\mathbf{t}$ is flattened to a vector of length $3N$. This allows the vector-valued measurement problem to be cast as a standard SBL problem, where the physical structure of the spherical harmonics expansion is preserved in the dictionary and the mapping between physical and matrix dimensions is explicit.

7.4 Antenna Simulation

7.5 Antenna Simulation

To validate the SBL-based near-field reconstruction approach described above, a set of synthetic antenna measurements was generated using MATLAB's Antenna Toolbox. Initially, an attempt was made to simulate more complex antenna structures in CST Microwave Studio. However, due to limitations in the student version of CST, specifically the lack of support for exporting radiation patterns in spherical coordinates, this path was not viable for generating high-resolution spherical near-field data.

An alternative approach was adopted by modeling simple canonical antennas—specifically, a half-wave dipole and a loop antenna—operating at 2.4 GHz using MATLAB's built-in functions. These provide well-defined and analytically tractable radiation characteristics, making them ideal for testing and validating reconstruction algorithms. The simulated electric field patterns were exported in spherical coordinates (r, θ, ϕ), with a resolution of $1°$ in both the polar (θ) and azimuthal (ϕ) directions.

Two distinct radial distances were considered for the field evaluation:

- $r = 0.5\lambda$: representing the reactive near-field region.
- $r = 5\lambda$: approximating the radiating far-field region.

This allowed for comparative analysis between near- and far-field reconstructions, as well as verification of the method's ability to capture the transition from near- to far-field behavior. The field components E_r, E_θ, and E_ϕ were sampled on a uniform spherical grid at each radius, forming the measurement vector $\mathbf{t}$ used in the SBL framework.
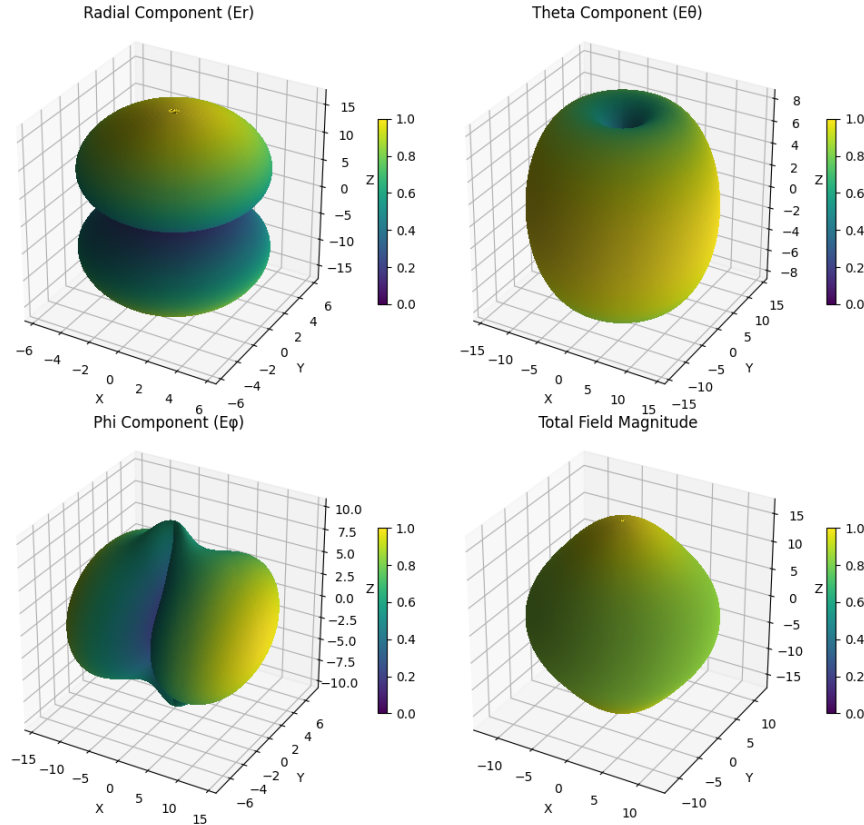
The results can be seen in figure 13.

Figure 13: The magnitude of the farfield radiation pattern of the dipole antenna.

7.6 Far-Field Pattern Expansion and Reconstruction

Initial experiments focused on the expansion of the simulated far-field radiation patterns into spherical vector wave harmonics, followed by their reconstruction using the derived coefficients. The electric field data obtained from MATLAB Antenna Toolbox at a radial distance of $r = 5\lambda$ (representing the far-field region) was used for this purpose.

The measurement matrix Φ (denoted as F in implementation) was constructed based on the theoretical expressions of the spherical vector wave functions $\mathbf{F}^{(c)}_{smn}$, evaluated at the corresponding wavenumber k and radius r. Subsequently, the wave coefficients $Q^{(3)}_{smn}$—corresponding to outward-propagating waves—were estimated using the Fast Marginal Likelihood Maximization algorithm.

To assess the accuracy of the expansion and reconstruction process, the original field $\mathbf{E}_{\mathrm{org}}$ was reconstructed via $\mathbf{E}_{\mathrm{rec}} = \Phi\mathbf{w}$, where $\mathbf{w}$ represents the estimated coefficient vector. The relative reconstruction error was quantified using the normalized mean squared error (MSE):

$$\mathrm{MSE}_{\mathrm{rel}} = \frac{\|\mathbf{E}_{\mathrm{rec}} - \mathbf{E}_{\mathrm{org}}\|}{\|\mathbf{E}_{\mathrm{org}}\|}.$$

These experiments were conducted using $N_{\mathrm{modes}} = 25$, resulting in a total of $D = 2N^2_{\mathrm{modes}} + 4N_{\mathrm{modes}} = 1350$ basis functions. The spatial resolution of the input field was systematically reduced by downsampling the angular sampling grid (θ, ϕ) with a factor k, leading to a varying number of measurement points N. The ratio $\delta = N/D$ was used as an independent variable to evaluate the reconstruction performance under different undersampling conditions.

Figure 14 shows the relative MSE as a function of δ for both the dipole and loop antennas. Notably, even at $\delta = 1$, where the number of measurements equals the number of unknowns, significant reconstruction errors are observed, particularly for the loop antenna. Ideally, a perfect reconstruction should be achievable in such a scenario, suggesting potential issues related to numerical inaccuracies, implementation flaws in the spherical vector wave functions $\mathbf{F}^{(c)}_{smn}$, or misinterpretations of the underlying electromagnetic theory.
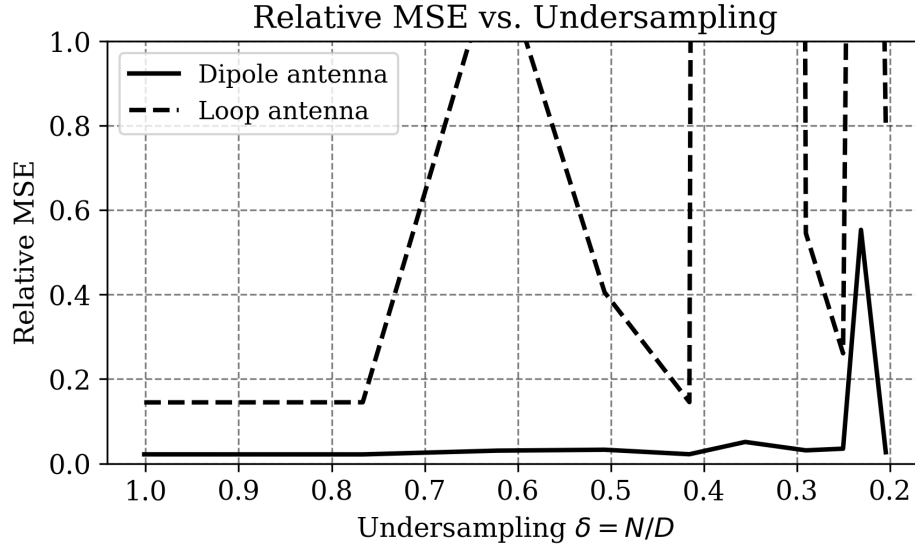

Figure 14: Relative reconstruction error (normalized MSE) as a function of the measurement-to-dimension ratio $\delta = N/D$ for dipole and loop antennas in the far-field configuration.

In contrast, the dipole antenna exhibits robust reconstruction performance even under substantial undersampling, indicating that its radiation pattern is well-represented within the sparse spherical harmonics domain.

However, the consistently higher reconstruction error for the loop antenna underscores the strong dependence of SBL performance on the specific radiation characteristics and topology of the source.

To further investigate the observed discrepancies and to validate the SBL results, a comparison was made between the coefficients obtained via the Fast Marginal Likelihood Algorithm and those computed through direct mode expansion using numerical integration, as theoretically described in Equation 28. For the dipole antenna, there was a notable agreement between the dominant coefficients identified by both methods, as illustrated in Figure 15. This consistency supports the validity of the SBL-based estimation approach.

However, for the loop antenna, shown in Figure 16, no such correspondence was observed. These results suggest either a fundamental difference in the sparsity structure of the loop antenna's field representation or unresolved inaccuracies in the implementation of the spherical wave basis functions. A fully validated wave mode expansion method would have been essential for clarifying these observations; unfortunately, such a method could not be reliably implemented due to the challenges previously described.
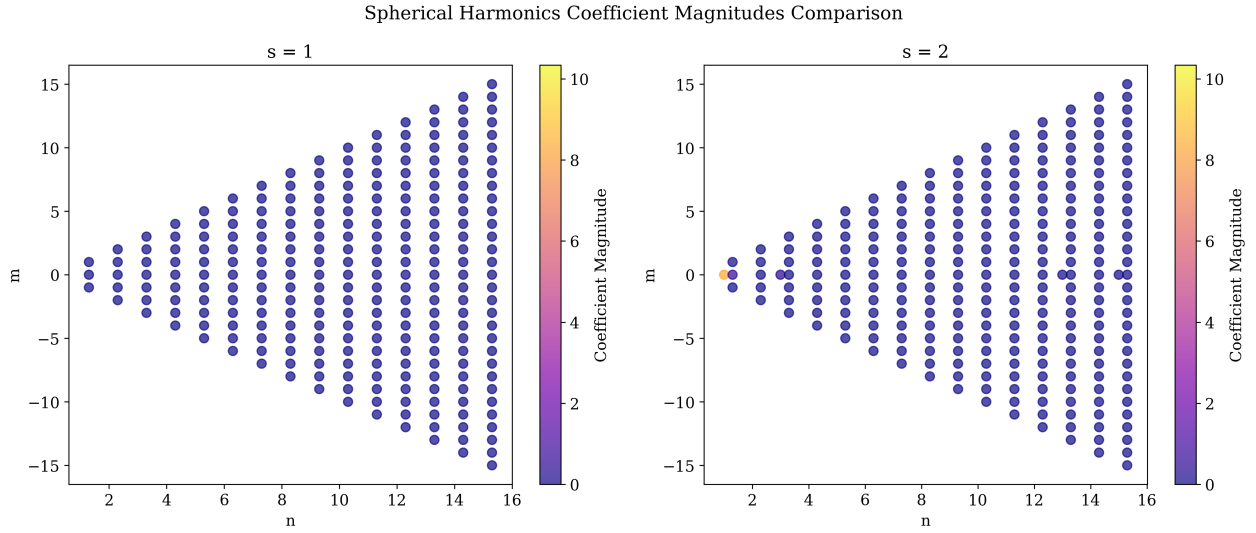

Figure 15: Comparison of wave coefficients for the dipole antenna: non-zero weights estimated by the Fast Marginal Likelihood Algorithm (left) versus significant coefficients obtained via direct integration (right).
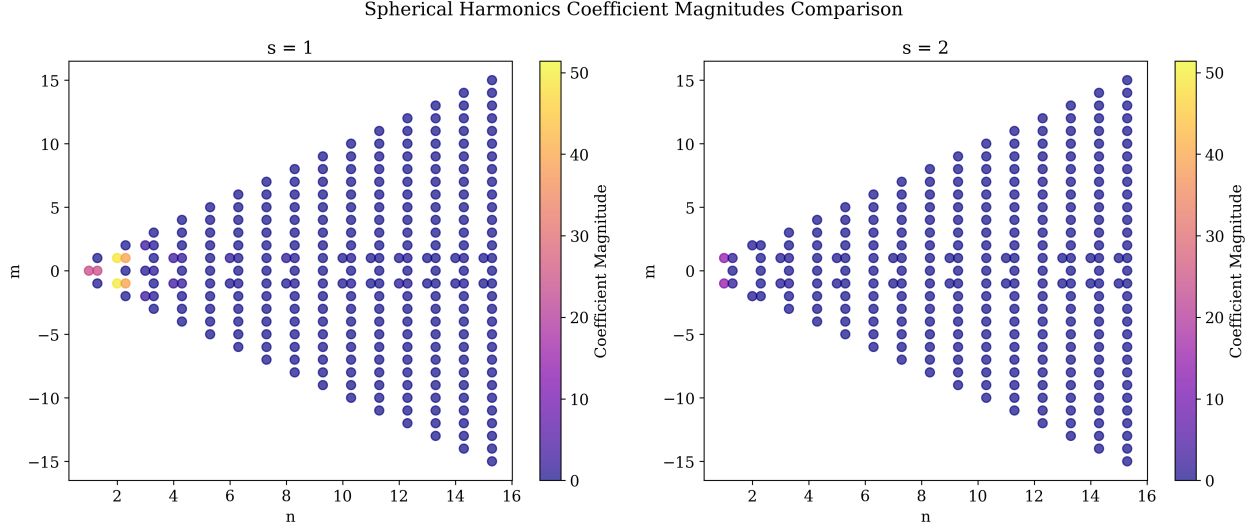
Figure 16: Comparison of wave coefficients for the loop antenna: non-zero weights estimated by the Fast Marginal Likelihood Algorithm (left) versus significant coefficients obtained via direct integration (right).

References

[1] J. E. Hansen, editor. *Spherical Near-Field Antenna Measurements*, volume 26 of *IEE Electromagnetic Waves Series*. Institution of Engineering and Technology, London, 1988. ISBN 978-0-86341-110-6. doi: 10.1049/PBEW026E. URL `https://digital-library.theiet.org/doi/book/10.1049/pbew026e`.

[2] Bernd Hofmann, Ole Neitz, and Thomas F. Eibert. On the minimum number of samples for sparse recovery in spherical antenna near-field measurements. *IEEE Transactions on Antennas and Propagation*, 67(12): 7597–7610, 2019. doi: 10.1109/TAP.2019.2935102.

[3] Alexander Lin, Andrew H. Song, Berkin Bilgic, and Demba Ba. Covariance-free sparse bayesian learning. *IEEE Transactions on Signal Processing*, 70:3818–3831, 2022. doi: 10.1109/TSP.2022.3186185.

[4] Arthur C. Ludwig. Near-field far-field transformations using spherical-wave expansions. *IEEE Transactions on Antennas and Propagation*, 19:214–220, 1971. URL `https://api.semanticscholar.org/CorpusID:15517872`.

[5] Dirk Slock. Sparse bayesian learning with stein's unbiased risk estimator based hyperparameter optimization. In *2022 56th Asilomar Conference on Signals, Systems, and Computers*, pages 857–861, 2022. doi: 10.1109/IEEECONF56349.2022.10052097.

[6] Ryan J. Tibshirani and Larry A. Wasserman. Stein ' s unbiased risk estimate statistical machine learning , spring 2015. 2015. URL `https://api.semanticscholar.org/CorpusID:39358969`.

[7] Michael E. Tipping and Anita C. Faul. Fast marginal likelihood maximisation for sparse bayesian models. In Christopher M. Bishop and Brendan J. Frey, editors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, volume R4 of *Proceedings of Machine Learning Research*, pages 276–283. PMLR, Jan 2003. URL `https://proceedings.mlr.press/r4/tipping03a.html`. Reissued by PMLR on 01 April 2021.

[8] D.P. Wipf and B.D. Rao. Sparse bayesian learning for basis selection. *IEEE Transactions on Signal Processing*, 52(8):2153–2164, 2004. doi: 10.1109/TSP.2004.831016.

[9] Qiuyun Zou and Hongwen Yang. A concise tutorial on approximate message passing, 2022. URL `https://arxiv.org/abs/2201.07487`.