Alternating Minimization Algorithm with Automatic Relevance Determination for Transmission Tomography under Poisson Noise

Yan Kaganovsky,† Shaobo Han,‡ Soysal Degirmenci, David G. Politte,‡ David J. Brady,*, Joseph A. O’Sullivan,† and Lawrence Carin,* §

Abstract. We propose a globally convergent alternating minimization (AM) algorithm for image reconstruction in transmission tomography, which extends automatic relevance determination (ARD) to Poisson noise models with Beer’s law. Sparsity of the imaged object is promoted in the pixel/voxel-differences domain by introducing additional latent variables, one for each pixel/voxel, and the latent variables are learned from the data using a hierarchical Bayesian model. Importantly, the proposed AM algorithm is free of any tuning parameters with image quality comparable to standard penalized-likelihood methods. Our algorithm exploits parallel line searches (one for each pixel/voxel) and is therefore feasible for large-scale problems; it considerably reduces the computational bottleneck of ARD associated with the posterior variances. Only smooth sparsity-promoting penalties are employed, thus allowing the use of standard gradient-based methods. Positivity constraints inherent in transmission tomography problems are also enforced. We demonstrate the performance of the proposed algorithm for x-ray computed tomography using synthetic and real-world datasets.

Code is available here: http://www.yan-kaganovsky.com/#code/c24bp

Key words. Alternating minimization, Automatic relevance determination, Transmission tomography, Poisson noise, X-ray CT.

1. Introduction. Tomographic image reconstruction is the process of estimating an object from measurements of its line integrals along different angles [1]. Tomographic reconstruction is an ill-posed problem [2] in the sense that multiple solutions exist that are consistent with the data. It is therefore often desirable to incorporate prior knowledge, which includes the statistical characteristics of the measured data, and properties that are expected of the image.

The images in transmission tomography represent attenuation per unit length of the impinging beam due to energy absorption inside the imaged object. Well known examples are x-ray computed tomography (CT) [3], and electron tomography [4]. The attenuation must be nonnegative, since a negative value corresponds to an increase in the energy of the exiting beam, which is non-physical. These images can often be well approximated by a sparse representation in some transform domain. In some applications, sparsity is present directly in the native image domain [5], but more commonly it is present in the pixel-difference domain [6] or in some transform domain, such as the wavelet, curvelet or shearlet transforms [7, 8].

There are generally two types of approaches for tomographic image reconstruction. The first type consists of one-shot algorithms, such as filtered back-projection [1] and its extensions, which rely on analytic formulas. These algorithms do not incorporate the type of prior knowledge mentioned above and also produce prominent artifacts when some of the data are missing. The second type consists of iterative algorithms, based on minimizing some cost
function. The latter enable incorporation of prior knowledge about the image, by adding penalties that promote sparsity in some representation and adding positivity constraints. In addition, iterative algorithms are far more robust to missing data.

Many transmission tomography problems involve data consisting of counts, e.g., the number of detected photons in x-ray CT, or the number of detected electrons in electron tomography. Therefore, at least some component of the measurement noise is modeled as independent Poisson random variables [6, 4]. In the x-ray medical imaging community, the Poisson likelihood model is standard [6] and provides the basis for a well known class of statistical iterative algorithms [6, 9, 10, 11]. In addition, the mean of each measurement is modeled according to Beer’s law [6], which is an inherent non-linearity in transmission data.

We develop a new sub-class of statistical iterative reconstruction algorithms, inspired by automatic relevance determination (ARD) [12, 13, 14]. It allows the incorporation of prior knowledge, including positivity of the image, Poisson statistics of the measured signals, Beer’s law and also any underlying sparse representations for the imaged object. What sets this new sub-class apart from prior art is the fact that it automatically learns the balance between data-fidelity and the penalty due to prior knowledge, thus avoiding the use of any tuning parameters and the difficulty in determining them. In addition, it also computes posterior variances, which can be used for adaptive sensing/experimental design, or for determining the system’s sensitivity to noise. To the best of our knowledge, the proposed algorithm is the first ARD-based algorithm for Poisson likelihoods that is both scalable to very large problems and that is also guaranteed to converge.

This work is mainly motivated by x-ray CT, used in medical and security applications [1], but the methods presented here can be applied more generally to any type of transmission tomography that involves count data.

1.1. Model. We consider a Poisson noise model for transmission tomography [6] given by

\[
p(y|x) = \prod_{i=1}^{n} p(y_i|x) = \prod_{i=1}^{n} \text{Poisson}[\eta_i \exp(-\phi_i^T x)],
\]

where \( y \in \mathbb{Z}^n \) are the measurements, \( \text{Poisson}[\lambda] \) denotes the Poisson univariate probability function with rate \( \lambda \), and the exponential function encompasses Beer’s law (a physical property of the signals [6]). The column vector \( x \in \mathbb{R}^{p \times 1} \) denotes the lexicographical ordering of the 2D/3D image, with each entry \( x_j \) representing the attenuation of the x-ray beam per unit length at the \( j \)th pixel/voxel. The vector \( \phi_i^T \in \mathbb{R}^{1 \times p} \) is the \( i \)th row of the system matrix \( \Phi \in \mathbb{R}^{n \times p} \). The measurement \( y_i \) corresponds to the \( i \)th source-detector pair and the operation \( \phi_i^T x \) represents a line integral over the image domain from source to detector (see Fig. 1). \( \eta_i \) is the mean number of photons that would have been measured using the \( i \)th source-detector pair if the imaged object was removed. We make the standard assumption that \( \eta_i \) was measured for all \( i \) in an independent calibration experiment and is known to us.

To construct \( \Phi \) we use the approach in [15], where an entry \( \phi_{ij} \) at the \( i \)th row and \( j \)th column of \( \Phi \) is equal to the length at which the line between the \( i \)th source-detector pair intersects the \( j \)th pixel/voxel (see Fig. 1). Each ray transverses at most \( O(p^{1/D}) \) pixels/voxels, where \( p \) is the total number of pixels/voxels and \( D = 2, 3 \) for 2D/3D images, respectively.
The total attenuation for the \( i \)th measurement is given by the line integral over the image along the path defined by the \( i \)th source-detector pair. Each line integral is approximated as \( \phi_{ij} \) where \( x \) are the pixel values and \( \phi_{ij} \) is the intersection length of the \( i \)th line with the \( j \)th pixel. Several measurements are collected simultaneously using several detectors. The source and detectors are rotated to collect many different views of the object.

Accordingly, each row \( \phi_i^T \) in the system matrix has only \( O(p^{1/D}) \) non-zero entries. We absorb a reference linear attenuation coefficient into \( \Phi \) so that \( x \) is dimensionless.

### 1.2. Background

The method proposed in this paper is an extension of automatic relevance determination (ARD). For convenience of readers unfamiliar with ARD, we first provide background to explain the potential advantages of ARD motivating this work, and discuss the key differences between ARD and other, more common, sparse estimation methods. Our aim here is to emphasize previous challenges in adapting ARD to the Poisson model in (1.1), challenges which we aim to resolve with the proposed method.

#### 1.2.1. Type I Methods

Statistical iterative methods for CT \([6,9,10,11]\) cast the problem of image reconstruction as a penalized maximum likelihood estimation problem, given by

\[
\mathbf{x}^* = \arg\max_{\mathbf{x}} [\log p(\mathbf{y}|\mathbf{x}) + \log p(\mathbf{x}|\beta)].
\]

We refer to (1.2) as the “Type I” estimation problem; \( \log p(\mathbf{y}|\mathbf{x}) \) is the log-likelihood function which represents the data-fit. For the Poisson model in (1.1), maximizing the log-likelihood is equivalent to minimizing the I-divergence \(^1\) between the actual measurements \( \mathbf{y} \) and the predicted measurements \( \eta \odot \exp(-\Phi \mathbf{x}) \) \([11]\), where \( \odot \) denotes an elementwise product. The expression \( \log p(\mathbf{x}|\beta) \) is a penalty that promotes solutions with desired properties, e.g., the \( \ell_1 \) penalty \([5]\) and the total variation (TV) penalty \([17,18]\); \( \beta \) is a tuning parameter that determines the trade-off between data-fit and the prior, or the bias/variance trade-off.

---

\(^1\)I-divergence between two vectors \( \mathbf{p}, \mathbf{q} \in \mathbb{R}_+^M \) is defined as \( I(\mathbf{p}|\mathbf{q}) = \sum_{j=1}^M p_j \log \left( \frac{p_j}{q_j} \right) - p_j + q_j \) \([16]\)
“best” value of $\beta$ is usually unknown and depends greatly on the imaged object at hand. To determine $\beta$, it is often required to solve (1.2) many times for different values of $\beta$, examine all reconstructed images and then select the value that best fits the needs of the application, based on some chosen figure of merit. For many common choices of $p(x|\beta)$ considered below, decreasing $\beta$ will result in a higher variance of the reconstructed image for different noise realizations, and in noisier images. Increasing $\beta$ will result in increased bias and artifacts (e.g., stair-case artifacts). For a given $\beta$, the solution in (1.2) can be interpreted as the maximum a posteriori (MAP) solution, i.e., the maximizing point of the posterior distribution, with the posterior distribution given by

$$p(x|y) = p(y|x)p(x)/p(y).$$

The $\ell_1$ and TV penalties are non-differentiable and do not allow the use of standard gradient-based methods for solving (1.2). An alternative to TV is the class of neighborhood roughness penalties (Gibbs priors) [10] which have the following form

$$-\log p(x|\beta) = \beta \sum_{\{j,k\} \in \Theta} w_{jk} \pi(x_j - x_k),$$

where $w_{jk}$ are positive weights inversely proportional to the distance between the $j$th and $k$th pixels/voxels, $\Theta$ is a set of ordered pairs $\{j,k\}$ defining a neighborhood system and $\pi(x)$ is chosen to be a positive, even, and twice continuously differentiable function. To preserve sharp edges in the image, $\pi(x)$ is often chosen such that $\pi(x) \propto |x|$ for large $|x|$ and can be viewed as a smooth version of the $\ell_1$ penalty. A common choice is the Huber-class due to Lange [10]

$$\pi(x) = \delta^2[|x/\delta| - \log(1 + |x/\delta|)],$$

which approaches the quadratic $x^2$ for $\delta \to \infty$ and the linear function $\delta |x|$ as $\delta \to 0$. This type of penalties allows the use of standard gradient-based methods but come with the price of having an additional tuning parameter $\delta$, so the search for the tuning parameters becomes even more cumbersome. Decreasing $\delta$ will result in noisier images but with sharper edges, and increasing $\delta$ will result in images which are less noisy but with over-smoothed edges.

### 1.2.2. Automatic Relevance Determination (ARD)/Type II Methods

A different approach to sparse estimation is “automatic relevance determination” (ARD) [12]. Originally, ARD has been proposed for Gaussian likelihoods with the prior [13]

$$p(x|\gamma) = N(x|0, \Gamma), \quad \Gamma = \text{Diag}(\gamma),$$

where $\text{Diag}(\gamma)$ denotes a diagonal matrix with main diagonal $\gamma$. In this approach sparsity is promoted by treating $x$ as latent variables and seeking the newly introduced hyperparameters $\gamma \in \mathbb{R}_+^p$ that maximize the marginal likelihood $p(y|\gamma)$

$$\gamma^* = \max_\gamma \log p(y|\gamma),$$

where $p(y|\gamma)$ is also called the “model evidence” or “Type-II” likelihood [14]. We refer to (1.7) as the “Type-II estimation problem.” Once $\gamma^*$ is found, the estimate for $x$ is given by the
mean of the posterior distribution \( p(x|y, \gamma^*) \). Interestingly, the solutions to (1.7) have many \( \gamma_j \approx 0 \) and the posterior distribution for the corresponding parameters \( x_j \) becomes highly concentrated around zero\(^2\). The mechanism that promotes sparsity of \( \gamma^* \) has been revealed by Wipf et al. \[19, 20\], with the analysis specifically tailored for Gaussian likelihoods.

The solution to the Type-II problem in (1.7) was originally addressed in \[13\] using an expectation-maximization (EM) algorithm \[21\]. In the E-step, the Gaussian posterior probability distribution \( p(x|y, \gamma) \) is computed for given hyperparameters \( \gamma \). In the M-step, \( \gamma \) are updated based on the mean and variances of the posterior distribution found in the E-step. As opposed to Type I methods, which focus on the mode (maximum) of the posterior distribution, Type-II (ARD) methods also compute the covariance matrix (during the E step), or at least the marginal variances, which provide additional information. ARD-based algorithms are therefore computationally more demanding than Type-I methods, mainly due to the expensive task of estimating the variances. Previous extensions of ARD \[20, 22\] require the same type of operations and rely on approximations to the variances \[22\] to maintain scalability.

Despite computational complications, the ARD approach can be quite rewarding and excel in hard or ill-posed problems, as evidenced in several recent applications \[23\]. A very significant advantage of ARD methods is their ability to automatically learn the balance between the data-fit and the degree of sparsity. This avoids the tuning of nuisance parameters, which is often non-intuitive, object-dependent, and requires many trials. The posterior variances are key in determining this trade-off. The variances can also be used for adaptive sensing/experimental design \[22\], which is outside the scope of this paper and will be the subject of future work.

### 1.2.3. Difficulties in Previous ARD-based Methods for Poisson Models.

The ARD-EM algorithm described in Sec. 1.2.2 \[13\] cannot be directly applied to the Poisson likelihood in (1.1) since the E-step is intractable due to the non-conjugacy\(^3\) of the prior in (1.6) and the likelihood. Although we are not aware of any previously published work on ARD for Poisson models with Beer’s law, a simple remedy to the non-conjugacy is the Laplace method, discussed in \[13\]. In that approach, the posterior is approximated during the E-step using a multivariate Gaussian distribution, with mean equal to the MAP solution and a covariance matrix equal to the inverse of the Hessian matrix evaluated at the MAP solution. However, the resulting “EM-like” algorithm is not based upon a consistent objective function, and is not guaranteed to converge. In addition, it requires inversions of the Hessian matrix with computational complexity \( O(p^3) \), where \( p \) is the total number of pixels/voxels. This is not practically feasible for x-ray CT, where \( p \) could be the order of \( 10^8 \) and higher. Instead, one can use more sophisticated methods \[22, 24, 25\] to indirectly estimate the diagonal of the inverse, i.e., the marginal variances, since only these are required for updating the hyperparameters \( \gamma \) \[13\]. Note however that the work in \[22, 24, 25\] does not consider Laplace approximations, so these methods have not been tested for this case. Irrespective of how the variances are computed, the “EM-like” algorithm with the Laplace approximation is unsatisfactory from a principled perspective due to the lack of an objective function and due to the lack of

\( ^2 \)by employing Bayes’ rule in (1.3), it is readily shown that if the prior is concentrated around \( x_i = 0 \) then the corresponding posterior distribution is also concentrated around \( x_i = 0 \).

\( ^3 \)a prior is said to be conjugate to the likelihood if both the prior and posterior probability distributions are in the same family. The posterior is then given in closed-form.
convergence guarantees. The objective is also needed to assess convergence in practice. In
addition, we are faced with conceptual difficulties, since the analysis that revealed the sparsity
promoting mechanism of ARD by Wipf et al. [14, 19, 20] is restricted to Gaussian likelihoods.

1.3. Contributions. The contributions of this paper are summarized as follows.
1. We present a new alternating minimization algorithm for transmission tomography,
which extends automatic relevance determination (ARD) to Poisson models with
Beer’s law. The proposed algorithm is free of any tuning parameters and also has
a consistent objective function that can be evaluated in a computationally efficient
way, in order to assess convergence in practice.
2. We reveal the sparsity-promoting mechanism in the proposed ARD-based algorithm
for Poisson likelihoods. This preserves insights similar to those gained from previous
work [14, 19, 20] for Gaussian likelihoods.
3. We derive a scalable AM algorithm for transmission tomography, that is well adapted
to parallel computing. We reduce the computational complexity considerably by sim-
plifying the algorithm into parallel line searches (for each pixel/voxel) with the scalar
mean and variance updates also done in parallel.
4. We present a convergence analysis for the scalable AM algorithm and establish guar-
antees for global convergence\(^4\). To the best of our knowledge, this is the first ARD-based
algorithm for Poisson models that has convergence guarantees.
5. We study the performance of the algorithm on synthetic and real-world x-ray datasets.

1.4. Outline of the Paper. The rest of this paper is organized as follows. In Sec. 2 we
present a general overview of the proposed AM algorithm, which extends ARD to Poisson
likelihoods with Beer’s law, and explains how sparsity is promoted. In Sec. 3 we present a
scalable AM algorithm based on separable surrogates for the objective function, which is well
adapted to parallel computing and is feasible for large-scale problems. In Sec. 4 we present a
detailed analysis of the convergence properties of the algorithm presented in Sec. 3. In Sec. 5
we make connections between the proposed algorithm and related previous methods. We also
present an alternative view of the proposed ARD method that further explains how sparsity is
promoted. In Sec. 6 we extend the proposed ARD framework to allow the modeling of sparsity in
overcomplete representations, while retaining the convergence properties studied in Sec. 4.
In Sec. 7 we study the performance of the proposed scalable AM algorithm on synthetic and
real-world datasets with a parallel implementation used in the experiments. Conclusions are
presented in Sec. 8.

2. Variational ARD.

2.1. Prior. In many applications of transmission tomography sparsity is not present di-
rectly in the native basis but in some transform domain. Therefore, we consider a generalized
prior (similar to Seeger and Nickisch [22]) to address these cases

\[
p(x | \gamma) = N(x | 0, (\Psi^T \Gamma^{-1} \Psi)^{-1}), \quad \Gamma = \text{Diag}(\gamma),
\]

\(^4\)An algorithm is said to be “globally convergent” if for arbitrary starting points the algorithm is guaranteed
to generate a sequence of points converging to a solution [26]. This does not imply a globally optimal solution.
where $\gamma \in \mathbb{R}_+^p$ are newly introduced hyperparameters, and we assume a transformation
\begin{equation}
(2.2) \quad s = \Psi x,
\end{equation}
where $\Psi$ is known and $s$ is approximately sparse. By choosing $\Psi = I$, the prior in (2.1) reduces to (1.6). Any scalable algorithm will rely on efficient matrix-vector multiplication $\Psi x$. This is achieved if $\Psi$ is sparse or has some special structure, such as discrete Fourier or wavelet transforms.

In this work we focus on neighborhood penalties, which are commonly used for x-ray CT, the application explored in Sec. 7. We consider $\Psi$ given by
\begin{equation}
(2.3) \quad \Psi_{jk} = \psi_{jk} = \begin{cases} 1 & \text{if } j = k \\ -1/N_{\max} & \text{if } \{j, k\} \in \Theta \\ 0 & \text{otherwise} \end{cases},
\end{equation}
where $\Theta$ is the set of ordered pairs $\{j, k\}$ defining a neighborhood system and $N_{\max}$ is the maximal number of neighbors for a pixel/voxel. An example is shown in Fig. 2. The neighborhood is usually chosen to be small enough such that $\Psi$ will be sparse with at most $N_{\max} + 1$ non-zero elements in each row. For the choice in (2.3), each $s_j$ in (2.2) is equal to the difference between the $j$th pixel/voxel and the average of its neighboring pixels/voxels, i.e., $s_j = x_j - \sum_{k: \{j, k\} \in \Theta} x_k/N_{\max}$. For pixels/voxels at the boundaries of the image domain, we assume additional neighbors outside the image domain with values set to zero (Dirichlet boundary conditions), such that the number of neighbors for each pixel/voxel is the same (see Fig. 2). These boundary conditions agree with physical constraints typical of many real experimental setups, where the attenuation $x$ is zero outside the region of interest. Note that sparsity of the above pixel/voxel differences implies piecewise smooth images.

2.2. Variational View of the EM Algorithm for ARD. We start with a known variational view [27] of the EM algorithm which is used to solve (1.7) for Gaussian noise models. Here we introduce some of the notation used throughout this work, and discuss the type of difficulties encountered when trying to extend ARD to non-Gaussian noise models.

We rewrite the evidence as
\begin{equation}
(2.4) \quad \log p(y | \gamma) = -\mathcal{F}[q(x), \gamma] + D_{KL}[q(x) || p(x|y, \gamma)],
\end{equation}
where
\begin{align}
(2.5) \quad & \mathcal{F}[q(x), \gamma] = \mathbb{E}_{q(x)} \log[q(x) / p(y, x | \gamma)], \\
(2.6) \quad & D_{KL}[q || p] = \mathbb{E}_{q(x)} \log[q(x) / p(x | y, \gamma)].
\end{align}
$q(x)$ is the proposed variational distribution and $p(x | y, \gamma)$ is the exact posterior, $\mathcal{F}$ is the free variational energy (FVE) or negative evidence lower bound (ELBO), $D_{KL}[q || p]$ is the Kullback-Leibler (KL) divergence, and $\mathbb{E}_q$ is the expected value with respect to $q(x)$. The EM algorithm can be viewed [27] as minimizing the free variational energy (FVE) function $\mathcal{F}$ in (2.5) (or maximizing the ELBO) by alternating between updating $q(x)$ in the E-step and updating $\gamma$ in the M-step, as summarized below
\begin{align}
(2.7) \quad & \text{E : } q^{(t+1)} = \arg \min_q \mathcal{F}[q(x), \gamma^{(t)}] \\
(2.8) \quad & \text{M : } \gamma^{(t+1)} = \arg \min_{\gamma} \mathcal{F}[q^{(t+1)}(x), \gamma].
\end{align}
In the E-step, the optimal solution with zero KL divergence is \( q^{(t+1)} = p(x|y, \gamma^{(t)}) \), i.e., the exact posterior based on the previous \( \gamma^{(t)} \) values, which can be computed directly from (1.3).

In the M-step, minimizing \( \mathcal{F} \) with respect to \( \gamma \) can only increase the KL term, since it is nonnegative and overall the evidence is also increased.

For non-Gaussian likelihoods and the prior in (2.1), the E step in (2.7) is intractable, i.e., the expression for \( p(x|y, \gamma^{(t)}) \) is unknown and its direct calculation according to (1.3) involves a high-dimensional integral that is prohibitive to compute. Instead, we are forced to restrict \( q(x) \) to a tractable family of distributions to approximate the posterior. However, this results in an algorithm that is no longer guaranteed to increase the evidence \[28\]. This occurs because the KL term in (2.4) is never zero and can either decrease or increase after updating \( \gamma \). Furthermore, a local maximum of the evidence with respect to \( \gamma \) is generally not a local minimum of \( \mathcal{F} \) (or a local maximum of the ELBO), except for degenerate cases \[28\].

Therefore, when considering the above approach to extend ARD, we immediately encounter a conceptual difficulty, since most previous theoretical analysis of ARD \[14, 19, 20\] is based on the maximization of the evidence and its specific closed-form expression for the Gaussian likelihood. It is not immediately clear how to extend the ARD framework to the Poisson noise model in (1.1) in a manner which will preserve the insights gained from \[14, 19, 20\] and will explain why sparsity is being promoted by the ARD algorithm.

Perhaps surprisingly, we shall show that the variational algorithm discussed above still preserves the principles of ARD, despite the fact that it is not based on maximizing the evidence, but rather on maximizing the ELBO (lower bound for the evidence). We call this framework “Variational ARD” (VARD).

2.3. Alternating-Minimization Algorithm for Variational ARD. As mentioned in Sec. 2.2, the true posterior distribution \( p(x|y, \gamma) \) corresponding to the Poisson model in (1.1) and the prior in (2.1) is intractable, so we cannot solve the E-step in (2.7) exactly, and an approximation to the posterior should be used instead. Perhaps the most common approach is to use a free-form factorized posterior distribution (mean field variational Bayes) \[29\], however this is also intractable here due to the lack of conditional conjugacy. Instead, we restrict the posterior to a parametric multivariate Gaussian form. In addition, in order to reduce the computational complexity, we restrict ourselves to factorized distributions, i.e.,

\[
\mathcal{D} \triangleq \{ h(x) \mid h(x) = \prod_{i=1}^{p} \mathcal{N}(x_i; m_i, v_i), \text{ for some } m_i, v_i, \ i = 1, ..., p \},
\]

where \( \mathcal{N}(x_i; m_i, v_i) \) denotes a normal univariate distribution with mean \( m_i \) and variance \( v_i \). We propose to modify the EM algorithm in (2.7)-(2.8) by adding the constraint \( q \in \mathcal{D} \) during the E-step in (2.7). The resulting algorithm is defined by the following two steps

\[
\text{B : } (\mathbf{m}^{(t+1)}, \mathbf{v}^{(t+1)}) = \arg \min_{m \succeq 0, v \succeq 0} \mathbb{E}_{q(x;m,v)}[-\log p(y|x)] + D_{KL}[q(x; m, v) || p(x|\gamma^{(t)})]
\]

\[
\text{F : } \gamma^{(t+1)} = \arg \min_{\gamma} D_{KL}[q(x; \mathbf{m}^{(t+1)}, \mathbf{v}^{(t+1)}) || p(x|\gamma)],
\]
which are repeated until convergence. Note that we have rewritten the FVE of (2.5) as
\[ F = \mathbb{E}_{q(x|m,v)}[-\log p(y|x)] + D_{KL}(q(x|m,v)||p(x|\gamma)). \]
The first term in (2.10) does not depend on \( \gamma \) and was therefore omitted in (2.11). In the backward (B) step in (2.10), we update the approximation to the posterior \( q \), which is reduced to only searching for the variational parameters \((m, v)\). In the forward (F) step in (2.11), we update the generative (forward) model by finding \( \gamma \). Note that due to the constraint \( q \in D \) in (2.9), this is no longer an EM algorithm, and therefore we distinguish between the “E-step” and the “B-step,” and similarly between the “M-step” and the “F-step.”

Note that the posterior mean \( m \) is the main object of interest. It provides an estimate for the attenuation coefficients, which must be nonnegative. Accordingly, a non-negativity constraint on \( m \) was added in (2.10). The function in (2.10) is only defined for positive \( v \) values and using the extended-value function we assume \( F = \infty \) for \( v_j \leq 0 \) for any \( j \).

The AM algorithm formed by repeating the updates in (2.10)–(2.11) will reduce the FVE at each iteration (ELBO is increased). Importantly, as shown below, the AM algorithm leads to near-sparse solutions, i.e., many of the marginal posterior distributions will be highly concentrated around zero in the transform space defined by (2.2). The roles of \( v \) and \( \gamma \) in promoting sparse solutions will be made clear in Sec. 2.4.

For the Poisson model in (1.1), the prior in (2.1) and the approximate posterior proposed in (2.9), the objective function in (2.10) can be written as (dropping iteration number)

\[
(2.12) \quad F = \underbrace{\mathbb{E}_{q(x|m,v)}[-\log p(y|x)]}_{F_1(m,v)} + \underbrace{D_{KL}(q(x|m,v)||p(x|\gamma))}_{F_2(m,\gamma) + F_3(v,\gamma)}
\]

\[
(2.13) \quad F_1(m,v) = \sum_{i=1}^{n} \left[ y_i \sum_{j=1}^{p} \phi_{ij} m_j + \eta_i \exp \left( - \sum_{j} \phi_{ij} m_j + \sum_{j} \phi_{ij}^2 v_j / 2 \right) \right]
\]

\[
(2.14) \quad F_2(m,\gamma) = \frac{1}{2} \sum_{k=1}^{p} \gamma_k^{-1} (\sum_{j=1}^{p} \psi_{kj} m_j)^2
\]

\[
(2.15) \quad F_3(v,\gamma) = \frac{1}{2} \sum_{k=1}^{p} \sum_{j=1}^{p} \gamma_k^{-1} \psi_{kj}^2 v_j - \frac{1}{2} \sum_{j=1}^{p} \log v_j + \frac{1}{2} \sum_{k=1}^{p} \log \gamma_k;
\]

where \( \phi_{ij} = (\phi_i)_j \) are the entries of the forward operator, as defined in (1.1), and \( \psi_{ij} = (\Psi)_{ij} \) are the entries of the \( \Psi \) matrix defined in (2.1). In deriving the last term in (2.15), we assumed that \( \Psi \) is square and invertible.\(^5\) We provide a generalization in Sec. 6 which removes this restriction, and applies to any \( \Psi \).

Recalling the discussion in Sec. 1.1 about the sparse structure of \( \phi_i \), we note that the objective function in (2.12)–(2.15) has a computational complexity of \( O(np^1/D) \)\(^6\) which is feasible to compute in order to assess convergence in practice. In contrast, the original ARD [13] or other extensions [20, 22] require \( O(p^3) \) operations to evaluate the objective which

\(^5\) the exact term is \( \log |(\Psi^T \Gamma^{-1} \Psi)|^{-1} = \log |\Gamma| + \text{Const} = \sum_k \log \gamma_k + \text{Const} \) with the equality based on the fact that \( \Psi \) is square and invertible.

\(^6\) for convenience we restate the definitions here: \( n \) is the number of measurements, \( p \) is the total number of pixels/voxels, and \( D = 2,3 \) for 2D/3D images.
is prohibitive for large scale problems where \( p \) is very large, so the objective can only be approximated, e.g., via the stochastic estimator of Papandreou and Yuille [25].

The solution to the F-step in (2.11) is obtained by solving \( \nabla_{\gamma} D_{KL}[q||p] = 0 \) which yields

\[
\gamma^{(t+1)} = [\Psi m^{(t+1)}] \odot [\Psi m^{(t+1)}] + [\Psi \odot \Psi] \nu^{(t+1)},
\]

where \( \odot \) denotes elementwise product. It is straightforward to verify that (2.16) is indeed a minimizing point of (2.11) (for finite \( \gamma \)). For fixed \( \gamma \), the FVE objective function is jointly convex with respect to \((m, \nu)\), which implies that any local minimum of the objective in (2.10) is also a global minimum. Note that the FVE function is not jointly convex with respect to \((m, \nu, \gamma)\), so local minima are possible. In a similar way, the original ARD-EM algorithm [13] is also only guaranteed to find a local maximum of the evidence [32].

2.4. Discussion. The B-step in (2.10) is well understood from a Bayesian variational inference perspective, and it is equivalent to minimizing the KL divergence between the proposed posterior \( q(x) \) and the true posterior \( p(x|y, \gamma) \), as described in Fig. 3(a).

However, the F-step in (2.11) is somewhat puzzling, as it is unclear why it promotes sparse solutions, especially since using a Gaussian prior for Type-I MAP estimation does not lead to sparse solutions. To explain how sparsity is promoted, consider the \( s \)-space defined in (2.2), where the image has a sparse representation. Note that the KL divergence in (2.11) is invariant under the transformation in (2.2). The prior and posterior distributions become

\[
p(s|\gamma) = \mathcal{N}(0, \gamma), \quad q(s) = \mathcal{N}(\mu, \Sigma), \quad \mu = \Psi m, \quad \Sigma = \Psi \text{Diag}(\nu) \Psi^T,
\]

where \( \Psi \) is assumed to be invertible. We next make use of the following representation for a factorized Student’s-t distribution [19]

\[
t(s|\nu) \triangleq \prod_{j=1}^{p} t(s_j|\nu) = \prod_{j=1}^{p} \max_{\gamma_j \geq 0} \mathcal{N}(s_j; 0, \gamma_j) \varphi(\gamma_j; \nu),
\]

where \( t(s_j|\nu) = \Gamma[(\nu+1)/2]((1+s_j^2/\nu)^{-(\nu+1)/2}\sqrt{\nu/\Gamma(\nu/2)} \) is a univariate Student’s-t distribution, and \( \varphi(\gamma_j; \nu) = \gamma_j^{\nu/2}\exp(-\nu/2\gamma_j) \). To obtain (2.18), one exploits Fenchel duality [33]

\[
\log t(\sqrt{s_j}|\nu) = \max_{\lambda_j \geq 0} \lambda_j s_j - f^*(\lambda_j),
\]

where \( f^*(\lambda_j) \) is the Fenchel dual of \( \log t(\sqrt{s_j}|\nu) \), and then uses the monotonically increasing transformation \( \lambda_j = -1/2\gamma_j \) to obtain \( \log t(s_j|\nu) = \max_{\gamma_j \geq 0} (-s_j/2\gamma_j + f^*(-1/2\gamma_j)) \).

Exponentiating both sides of the dual representation for \( \log t(s_j|\nu) \) one obtains (2.18) [19]. For \( \nu \to 0 \), \( t(s_j|\nu) \propto 1/|s_j| \) with sharp spikes at \( s_j = 0 \). Note in (2.18) that \( \nu \to 0 \) implies \( \varphi \to 1 \), and the multivariate t-distribution \( t(s|\nu) \) becomes an envelope for the ARD prior \( p(s|\gamma) \) in (2.17). This is illustrated in Fig. 3(b) using a 2D example with \( s \in \mathbb{R}^2 \), which depicts the proposed posterior \( q(s) \) and the prior \( p(s|\gamma) \), with the latter “trapped” inside the level sets of \( t(s|\nu) \) due to (2.18). One can see in Fig. 3(b) how selecting the widths \( \gamma \) of the Gaussian prior distribution to minimize the KL divergence between \( q(s) \) and \( p(s|\gamma) \) promotes the shrinking of \( \gamma_k \) for some \( k \) (minimizing KL divergence between the two distributions is illustrated as maximizing the overlap between the level-set ellipsoids corresponding to the majority of the probability mass of these distributions). In the example of Fig. 3, the posterior has a much lower mean and variance along \( s_2 \) than along \( s_1 \), and the F-step will further shrink \( \gamma_2 \). The corresponding parameter \( s_2 \) will be shrunk in the
sense that the marginal prior $p(s_2|\gamma_2) = \mathcal{N}(s_2; 0, \gamma_2)$ will be more concentrated around zero, and so will the true posterior $p(s_2|y)$ (see (1.3)). In the following B-step, the approximate posterior $q$ will also concentrate further around zero due to the zero-forcing property of the KL divergence \[34\] (recall that in the B-step we minimize KL-divergence between $q$ and the true posterior). Note that repeating the B and F steps is required to obtain sparse solutions.

The description of the mechanism responsible for promoting sparsity was inspired by the work of Wipf et al.\[19\], which uses similar arguments; however, that work relies entirely on the evidence function $p(y|\gamma)$ in (2.4) since a Gaussian noise model was considered. In contrast, our proposed approach is based on the free variational energy (FVE) function and applies more generally, even to cases where the expression for the evidence is unknown and an algorithm that increases the evidence is unavailable. This is important for non-Gaussian noise models, for which the evidence function is intractable, so we are forced to work with the FVE function instead. Another important difference between our work and that of Wipf et al.\[19\] is that we show how sparsity is promoted during the iterations of the algorithm, whereas in \[19\] it has been shown that a solution to (1.7) is sparse. Note also that the work in \[19\] considers the original ARD prior in (1.6), whereas we consider the more general case in (2.1).

3. Parallel AM Algorithm. Here we present a modified AM algorithm, which uses an alternative B-step. The rationale behind the modified algorithm lies in decoupling the variational parameters $m_j, v_j$ of each pixel/voxel. This reduces the algorithm into simultaneous simple tasks that can be computed in parallel, leading to high scalability. First, note that (2.13) and (2.14) are not additively separable with respect to the individual parameters $m_j, v_j$. We shall therefore introduce a surrogate function, making use of the following lemma.

![Figure 3](image-url) Description of the AM algorithm proposed in (2.10)–(2.11). (a) Backward step: level sets for the likelihood $p(y|x)$ (red curve), prior $p(x)$ (blue curve) and the proposed posterior $q(x)$ (black curve); true posterior $p(x|y)$ is concentrated in the orange region; proposed posterior $q(x)$ is chosen to minimize the KL divergence with $p(x|y)$. (b) Forward step: a level set for the student's t distribution (dashed black curves) forming an envelope for the level set of the zero mean Gaussian prior $p(s)$ (blue curves); $p(s|\gamma^{(1)})$ and $p(s|\gamma^{(2)})$ correspond to two different choices for $\gamma$ where $p(s|\gamma^{(2)})$ assigns equal variance along $s_1$ and $s_2$ and $p(s|\gamma^{(1)})$ assigns small variance along $s_2$ and large variance along $s_1$. Choosing $p(s|\gamma^{(1)})$ maximizes the overlap between the prior $p(s)$ and the proposed posterior $q(s)$ (leading to lower KL divergence). $s_2$ is shrunk in the sense that its exact and proposed posterior distributions are further concentrated around smaller values due to the concentration of the prior about zero. Note that Fig. (a) is in the $x$ domain whereas Fig. (b) is in the $s$ domain.
Lemma 3.1 (The convex decomposition lemma [11]). Let $f(x)$ be a convex function with $x \in \mathbb{R}^p$ and let $\hat{x} \in \mathbb{R}^p$ be a reference point. Let $r_j \geq 0$ such that $\sum r_j = 1$. Then

\begin{equation}
(3.1) \quad f(x) \leq \sum_j r_j f(\hat{x} + \frac{x_j - \hat{x}_j}{r_j} e_j),
\end{equation}

where $e_j$ denotes a column vector with all zeros except 1 at the $j$th entry. **Equality** in (3.1) is obtained iff $x_j = \hat{x}_j \forall j$.

**Proof.** The proof readily follows from Jensen’s inequality [33]

\begin{equation}
(3.2) \quad f(x) = f(\hat{x} + \sum_j (x_j - \hat{x}_j)e_j) = f(\hat{x} + \sum_j r_j \frac{x_j - \hat{x}_j}{r_j} e_j) \leq \sum_j r_j f(\hat{x} + \frac{x_j - \hat{x}_j}{r_j} e_j),
\end{equation}

where we added and subtracted a reference point $\hat{x}$ and then applied Jensen’s inequality to $g(\sum_j r_j [\nu_j/r_j])$ where $\nu_j \triangleq (x_j - \hat{x}_j)e_j$, and $g(x) \triangleq f(\hat{x} + x)$. Equality is obtained iff $\nu_j/r_j$ is independent of $j$, or equivalently, iff $x_j = \hat{x}_j \forall j$.

Before we proceed, we make several definitions to simplify the following derivations.

\begin{equation}
(3.3) \quad p_i^{(t)} \triangleq \sum_j \phi_{ij} m_j^{(t)}, \quad \tilde{p}_i^{(t)} \triangleq \sum_j \phi_{ij} r_j^{(t)},
\end{equation}

\begin{equation}
(3.4) \quad \mu_i^{(t)} \triangleq \mathbb{E}_{q(x,m^{(t)},\nu^{(t)})} [\eta_i \exp(-\phi_i^T x)] = \eta_i \exp(p_i^{(t)}/2) \exp(-p_i^{(t)}),
\end{equation}

\begin{equation}
(3.5) \quad b_j^{(t)} \triangleq \sum_i \phi_{ij} y_i, \quad \tilde{b}_j^{(t)} \triangleq \sum_i \phi_{ij} \mu_i^{(t)}, \quad \hat{b}_j^{(t)} \triangleq \sum_i \phi_{ij}^2 \mu_i^{(t)}/2,
\end{equation}

where $y_i$, $\phi_i$ and $\eta_i$ are defined in (1.1). To distinguish between quantities associated with the mean and variance, we denote any quantities associated with the variance using a tilde. In (3.3), $p_i^{(t)}$ is the projection of the posterior mean $m^{(t)}$ along the $i$th line, which we shall call “mean-type” projection; $\tilde{p}_i^{(t)}$ is a projection of the posterior variance $\nu^{(t)}$ along the $i$th line using an elementwise squared forward operator, which we term a “variance-type” projection. In (3.4), $\mu_i^{(t)}$ is the predicted Poisson rate for the $i$th measurement, based on the expectation with respect to posterior distribution $q$ computed at iteration $t$. In (3.5), $b_j^{(t)}$ is the back-projection of measurements $y$ to the $j$th pixel; $\hat{b}_j^{(t)}$ and $\hat{b}_j^{(t)}$ are the back-projections of the predicted Poisson rate in (3.4) to the $j$th pixel, using the backward (adjoint) operator and the squared backward operator, respectively, which we call “mean-type” and “variance-type” backprojections.

By noting that $\mathcal{F}_1$ in (2.13) is jointly convex with respect to $(m,v)$, and applying Lemma 3.1 with $\hat{x} = (m^{(t)}, \nu^{(t)})$, we obtain the bound $\mathcal{F}_1 \leq S_1^{(t)}(m,v)$ at iteration $t$ with $S_1^{(t)}$ defined as

\begin{equation}
(3.6) \quad S_1^{(t)}(m,v) = \sum_i \sum_j y_i \phi_{ij} m_j + \mu_i^{(t)} r_{ij} \exp \left[- \frac{\phi_{ij}}{r_{ij}} (m_j - m_j^{(t)}) \right] + \mu_i^{(t)} \hat{r}_{ij} \exp \left[ \frac{\phi_{ij}^2}{r_{ij}} (v_j - v_j^{(t)})/2 \right],
\end{equation}

where we made use of the definitions in (3.3)–(3.5). Note that (3.6) is separable with respect to the components $m_j$ and $v_j$. We are free to choose $r_{ij}, \hat{r}_{ij}$ as long as the conditions of
Lemma 3.1 are satisfied. We follow O’Sullivan and Benac [11] to apply a minor (but useful) extension of the lemma, where we introduce a dummy variable denoted \( m_0 \) with a corresponding weight \( r_0 \geq 0 \). We set \( m_0 = 0 \) so that the resulting algorithm does not involve \( m_0 \) in any way. This modification leads to the (relaxed) requirement that

\[
\sum_j r_{ij} + \sum_j \tilde{r}_{ij} \leq 1
\]

instead of equality. The above relaxation allows us to make the following choice

\[
\begin{align*}
  r_{ij} &= \frac{\phi_{ij}}{Z_1}, \\
  \tilde{r}_{ij} &= \frac{\phi_{ij}^2}{2Z_1}, \\
  Z_1 &= \max_i \sum_j \phi_{ij} + \phi_{ij}^2/2,
\end{align*}
\]

which satisfies the conditions of the extended lemma. Substituting (3.8) into (3.6), and using the definitions in Eq. (3.5), we obtain

\[
S_1^{(t)}(m, v) = \sum_j b_j^t m_j + \frac{b_j^t}{Z_1} \exp \left[ -Z_1(m_j - m_j^{(t)}) \right] + \frac{\tilde{b}_j^t}{Z_1} \exp \left[ Z_1(v_j - v_j^{(t)}) \right].
\]

The particular choice in (3.8) is very useful since the sum over \( i \) in (3.6) needs to be done only once instead of repeating it for each update of \( m_j \) during the search for the minimum of \( S_1 \).

We derive a separable bound for (2.14), \( F_2 \leq S_2^{(t)}(m, v) \), by noting that \( x^2 \) is a convex function and using lemma 3.1 again to obtain

\[
S_2^{(t)}(m, v) = \frac{1}{2} \sum_k \gamma_k^{(t)} \sum_j s_{kj} \left[ (d_k^{(t)})^2 + 2d_k^{(t)} \psi_{kj}(m_j - m_j^{(t)}) + \frac{\psi_{kj}^2}{s_{kj}}(m_j - m_j^{(t)})^2 \right],
\]

where \( d_k^{(t)} \triangleq \sum_j \psi_{kj} m_j^{(t)} \) (also defined in (3.13)). Similarly to (3.7), we require \( \sum_j s_{kj} \leq 1 \) using the dummy variable extension. One choice for \( s_{kj} \) is

\[
s_{kj} = \frac{\psi_{kj}}{Z_2}, \quad Z_2 = \max_k \sum_j |\psi_{kj}|.
\]

Substituting (3.11) into (3.10), we obtain

\[
S_2^{(t)}(m, v) = h^{(t)} + \sum_j [f_j^{(t)}(m_j - m_j^{(t)}) + g_j^{(t)}(m_j - m_j^{(t)})^2],
\]

where we introduced the following definitions

\[
\begin{align*}
  d_k^{(t)} &\triangleq \sum_j \psi_{kj} m_j^{(t)}, & f_j^{(t)} &\triangleq \sum_k \psi_{kj} d_k^{(t)}/\gamma_k^{(t)}, \\
  g_j^{(t)} &\triangleq Z_2 \sum_k |\psi_{kj}|/2\gamma_k^{(t)}, & h^{(t)} &\triangleq \sum_k (d_k^{(t)})^2/2\gamma_k^{(t)}, \\
  \xi_j^{(t)} &\triangleq \sum_k \psi_{kj}^2/\gamma_k^{(t)}.
\end{align*}
\]
Combining (3.20) (the latter is already separable), we obtain a surrogate function $S^{(t)}$ that is a global majorizer of $F$, i.e., $F(m, \gamma) \leq S^{(t)}(m, v)$, given by

$$S^{(t)}(m, v) = \sum_j S_{m_j}^{(t)}(m_j) + \sum_j S_v^{(t)}(v_j),$$

$$S_{m_j}^{(t)}(m_j) = b_j^0 m_j + \frac{b_j^{(t)}}{Z_1} \exp \left[ -Z_1(m_j - m_j^{(t)}) \right] + f_j^{(t)}(m_j - m_j^{(t)}) + g_j^{(t)}(m_j - m_j^{(t)})^2,$$

$$S_v^{(t)}(v_j) = \frac{\tilde{v}_j^{(t)}}{Z_1} \exp \left[ Z_1(v_j - v_j^{(t)}) \right] + \xi_j^{(t)} v_j / 2 \log v_j / 2.$$

Note that the surrogate function $S$ in (3.16) is separable with respect to both the components of $m$ and $v$. This yields a modified B-step where $S$ is minimized by simultaneous (parallel) line search procedures, each with respect to a different component $m_j$ or $v_j$. The iterations are defined as

$$m_j^{(t+1)} = \arg\min_{m_j \geq 0} S_{m_j}^{(t)}(m_j),$$

$$v_j^{(t+1)} = \arg\min_{v_j \geq 0} S_v^{(t)}(v_j).$$

Combining the modified B-step in (3.19)–(3.20) with the F-step in (2.16), we obtain the parallel VARD algorithm described in Algorithm 1. Each iteration requires a “mean-type”

---

**Algorithm 1**: PAR-VARD (VARD with Separable Surrogates)

1. Initialize $m^{(1)}, v^{(1)}, \gamma^{(1)}$
2. Compute $b^0$ in (3.5) % backprojection of data %
3. for $t=1$ to $N$ do % AM iterations %
4. $p^{(t)} \leftarrow \Phi m^{(t)}$ %“mean-type ” projections %
5. $\tilde{p}^{(t)} \leftarrow (\Phi \otimes \Phi)p^{(t)}$ %“variance-type” projections %
6. $\mu^{(t)} \leftarrow \eta \odot \exp(\tilde{p}^{(t)}/2) \odot \exp(-p^{(t)})$
7. Compute $g^{(t)}, \xi^{(t)}$ and $f^{(t)}$ defined in (3.13)–(3.15)
8. $\tilde{b}^{(t)} \leftarrow \Phi^T \mu^{(t)}$ % “mean-type” backprojections of $\mu^{(t)}$ %
9. $\tilde{b}^{(t)} \leftarrow (\Phi \otimes \Phi)^T \mu^{(t)}$ % “variance-type” backprojections of $\mu^{(t)}$ %
10. for $j=1$ to $p$ do % executed in Parallel %
11. $m_j^{(t+1)} \leftarrow \arg\min_{m_j \geq 0} S_{m_j}^{(t)}(m_j; \tilde{b}_j^{(t)}, f_j^{(t)}, g_j^{(t)})$ % see (3.17) %
12. $v_j^{(t+1)} \leftarrow \arg\min_{v_j \geq 0} S_v^{(t)}(v_j; \tilde{v}_j^{(t)}, \xi_j^{(t)})$ % see (3.18) %
13. $\gamma^{(t+1)} \leftarrow [\Psi m^{(t)}] \odot [\Psi m^{(t)}] + [\Psi \odot \Psi]p^{(t)}$ % see (2.16) %
forward-projection \( p(t) \) and a “variance type” forward-projection \( \tilde{p}(t) \) (lines 4-5 in Algorithm 1) which can be computed in parallel. It also requires a “mean type” backprojection \( b(t) \) and a “variance-type” backprojection \( \tilde{b}(t) \) (lines 8-9 in Algorithm 1), which can also be computed in parallel. Each iteration also requires two line searches, one for the mean components \( m_j \) and one for the variance components \( v_j \) (lines 11-12 in Algorithm 1). The line searches for all pixels/voxels can be done in parallel. Note that despite the latter parallelization, \( b(t) \) and \( \tilde{b}(t) \) provide information about both the mean and variance at the previous iteration which is shared between the mean and variance updates in (3.19)–(3.20) (lines 11-12 in Algorithm 1). Also note that these subiterations for \((m, v)\) are themselves iterations of another alternating minimization algorithm (lines 4-12 of Algorithm 1). Here we chose to use only one subiteration, but any number can be used instead.

A considerable advantage of this algorithm is that it allows fast parallel computing on multi-core computers, or on dedicated hardware such as field-programmable gate arrays (FPGA). Another advantage is that the solution to the constrained problem in (3.19) can be obtained by first solving the unconstrained problem using simple gradient-based methods; if \( m_j < 0 \) then the solution to the constrained problem must be \( m_j = 0 \), otherwise the solution is the same. This simple thresholding for \( m_j \) leads to the optimal solution only in one-dimensional minimization problems, and the same procedure for the original B-step in (2.10) would lead to sub-optimal solutions or even lack of convergence.

The computational complexity of a forward/backprojection is \( O(np^{1/D}) \), where \( n \) is the number of measurements, \( p \) is the total number of pixels/voxels and \( D = 2, 3 \) for 2D/3D images, respectively (recall the discussion about the sparsity of \( \Phi \) in Sec. 1.1). The computational complexity of \( \Psi m \) is \( O(p(N_{\text{max}} + 1)) \), where \( N_{\text{max}} \) is the maximal number of neighboring pixels/voxels defined by the prior (recall the discussion on the sparsity of \( \Psi \) in Sec. 2.1). Typically, \( N_{\text{max}} \approx 3^D - 1, n \geq p, \) and \( p^{1/D} \gg N_{\text{max}} \), so the forward/backprojection cost dominants the cost of \( \Psi m \). The details regarding the implementations of the line searches (lines 11-12 in Algorithm 1) are given in Appendix B.

4. Optimality Conditions and Convergence Analysis. First we provide the necessary optimality conditions for the minimization problems presented in Secs. 2–3 as well as a few definitions in order to simplify the following analysis. We then present a detailed study of the convergence properties of the parallel AM algorithm derived in Sec. 3 (Algorithm 1). We shall refer to this algorithm as “PAR-VARD” (parallel VARD).

4.1. Necessary Optimality Conditions. Proposition 4.1.

1. The necessary Karush-Kuhn-Tucker (KKT) optimality conditions [33] for \((m, v)\) to be a solution to (2.10) are given by

\[
\begin{align*}
 b_j^\ast &+ \sum_k \psi_{kj} d_k / \gamma_k - b_j \begin{cases} = 0 & \forall j \text{ s.t. } m_j > 0 \\
 \geq 0 & \forall j \text{ s.t. } m_j = 0 \end{cases}, \\
 \tilde{b} & = \lambda_j^{(t)} - 1/v_j, \quad \forall j,
\end{align*}
\]

where \( b_j^\ast, b_j, \tilde{b}_j \) are defined by (3.5) with \( m_j^{(t+1)} \) and \( v_j^{(t+1)} \) replaced by \( m_j \) and \( v_j \),
respectively; $d_k$ is defined by $d_k^{(t)}$ in (3.13) with $m_j^{(t)}$ replaced by $m_j$; $\xi_j^{(t)}$ is defined in (3.15).

2. The necessary KKT condition for $\gamma$ to be a solution to (2.11) is given by

$$\gamma = (\Psi m^{(t+1)}) \odot (\Psi m^{(t+1)}) + (\Psi \odot \Psi) v^{(t+1)}.$$  \hfill (4.3)

3. The necessary KKT conditions for $(m, v, \gamma)$ to be a minimizer of the FVE objective in (2.12)-(2.15) subject to $m_\geq 0$ are given by

$$b_j^y + f_j - b_j \begin{cases} = 0 \quad \forall j \text{ s.t. } m_j > 0, \\ \geq 0 \quad \forall j \text{ s.t. } m_j = 0 \end{cases}, \quad \hfill (4.4)$$

$$\tilde{b} = \xi_j - 1/v_j, \quad \forall j, \quad \hfill (4.5)$$

$$\gamma = (\Psi m) \odot (\Psi m) + (\Psi \odot \Psi) v, \quad \hfill (4.6)$$

where $\xi_j$ is defined as $\xi_j^{(t)}$ in (3.15) with $\gamma^{(t)}$ replaced by $\gamma_j$; $f_j$ is defined as $f_j^{(t)}$ in (3.13) with $d_k^{(t)}$ and $\gamma_k^{(t)}$ replaced by $d_k$ and $\gamma_k$, respectively (see also definitions after (4.2)).

**Proof.** Follows directly from the KKT conditions $\nabla m_j F \geq 0 \forall j \text{ s.t. } m_j = 0, \nabla m_j F = 0 \forall j \text{ s.t. } m_j > 0, \nabla v F = 0, \nabla_{\gamma} F = 0$, and by plugging in (2.12)-(2.15). For (1) substitute $\gamma = \gamma^{(t)}$ and use only KKT conditions for $(m, v)$; For (2) substitute $(m, v) = (m^{(t+1)}, v^{(t+1)})$ and use KKT condition for $\gamma$. \hfill \square

**Remark 1:** note that $v_j > 0$ due to the $-\log v_j$ term in (2.15). **Remark 2:** note that (4.3) is the F-step update in (2.16) so it is satisfied at the end of each iteration of PAR-VARD.

**Proposition 4.2.**

1. The necessary KKT conditions for $m_j$ to be a solution to (3.19) are

$$b_j^y - b_j^{(t)} \exp \left[ - Z_1(m_j - m_j^{(t)}) \right] + f_j^{(t)} + 2g_j^{(t)} (m_j - m_j^{(t)}) \begin{cases} = 0 \quad \forall j \text{ s.t. } m_j > 0, \\ \geq 0 \quad \forall j \text{ s.t. } m_j = 0 \end{cases}, \quad \hfill (4.7)$$

where $b_j^{(t)}, b_j^{(t)}, f_j^{(t)}, g_j^{(t)}$ and $Z_1$ are defined in (3.5), (3.13), (3.14) and (3.8), respectively.

2. The necessary KKT condition for $v_j$ to be a solution to (3.20) is

$$\tilde{b}_j^{(t)} \exp \left[ Z_1(v_j - v_j^{(t)}) \right] + \xi_j^{(t)}/2 - 1/2v_j = 0, \quad \hfill (4.8)$$

where $\tilde{b}_j^{(t)}, \xi_j^{(t)}$ are defined in (3.5) and (3.15).

**Proof.** Conditions (4.7)-(4.8) follow directly from the KKT conditions: $\nabla m_S m_j \geq 0 \forall j \text{ s.t. } m_j = 0, \nabla m S m_j = 0 \forall j \text{ s.t. } m_j > 0, \nabla v S v_j = 0 \forall j$, and by plugging in (3.17)-(3.18). \hfill \square

**Remark:** Note that $v_j > 0$ due to the $-\log v_j$ term in (3.18).

**4.2. Definitions.** To simplify the convergence study in Sec. 4.3, we introduce a few definitions concerning the solutions to (3.19)-(3.20).
Definition 4.3. Let \( \tilde{m}_j^{(t)} \) be a solution to (3.19) but without the non-negativity constraint. The corresponding solution to (3.19) with the non-negativity constraint can be expressed as

\[
m_j^{(t+1)} = \left[ \tilde{m}_j^{(t)} - \frac{1}{Z_1} \log \left( \frac{\beta_j^{(t)}}{\hat{b}_j^{(t)}} \right) \right]_+,
\]

where \([x]_+ = \max[0, x]\) and \(\beta_j^{(t)}\) is defined as

\[
\beta_j^{(t)} = \hat{b}_j^t + f_j^{(t)} + 2g_j^{(t)} (\tilde{m}_j^{(t)} - m_j^{(t)}).
\]

**Remark:** note from (3.5) that \(\hat{b}_j^t \geq 0\), and from (4.7) and (4.10) it follows that \(\beta_j^{(t)} \geq 0\), so the logarithm in (4.9) is well defined.

Definition 4.4. The solution to (3.19) can alternatively be expressed as

\[
m_j^{(t+1)} = m_j^{(t)} - \frac{1}{Z_j^{(t)}} \log \left( \frac{\tilde{m}_j^{(t)}}{\hat{b}_j^{(t)}} \right),
\]

where we define

\[
Z_j^{(t)} \triangleq \begin{cases} 
Z_1 & \text{for } \tilde{m}_j^{(t)} \geq 0 \\
\frac{1}{m_j^{(t)}} \log \left( \frac{\tilde{m}_j^{(t)}}{\hat{b}_j^{(t)}} \right) & \text{for } \tilde{m}_j^{(t)} < 0, \text{ and } m_j^{(t)} > 0 \\
\infty & \text{for } \tilde{m}_j^{(t)} < 0 \text{ and } m_j^{(t)} = 0.
\end{cases}
\]

**Remark 1:** introducing \(Z_j^{(t)}\) in (4.11) has the same effect as thresholding in (4.9). **Remark 2:** note that since \(m^{(t+1)} \geq \tilde{m}\) we have \(Z_j^{(t)} \geq Z_1\). We also introduce a similar definition for the variance.

Definition 4.5. The solution to (3.20) can be expressed as

\[
v_j^{(t+1)} = v_j^{(t)} + \frac{1}{Z_1} \log \left( \frac{\tilde{b}_j^{(t)}}{\hat{b}_j^{(t)}} \right),
\]

where \(\tilde{b}_j^{(t)}\) is defined as

\[
\tilde{b}_j^{(t)} \triangleq \frac{1}{2v_j^{(t+1)} - \xi_j^{(t+1)}/2}.
\]

**Remark:** since \(\tilde{b}_j^{(t)} \geq 0\) and \(\xi_j^{(t)} \geq 0\), it follows from (4.8) that \(v_j^{(t+1)} > 0\).

Definition 4.6. The I-divergence between two vectors \(p, q \in \mathbb{R}_+^M\) is defined as

\[
I(p||q) = \sum_{j=1}^{M} p_j \log \left( \frac{p_j}{q_j} \right) - p_j + q_j.
\]

We define \(\log(0^+0) = 0\). **Remark:** the I-divergence is nonnegative.

Definition 4.7. The Itakura-Saito (IS) divergence between two vectors \(p, q \in \mathbb{R}_+^M\) is defined as

\[
D_{IS}(p||q) = \sum_k \left[ \log(p_k/q_k) + q_k/p_k - 1 \right].
\]

We define \(\log(0^+0) = 0\), and \(0^+0 = 1\). **Remark:** \(D_{IS}\) is nonnegative.
4.3. Convergence Analysis. For simplicity we shall denote $\mathcal{F}^{(t)} \triangleq \mathcal{F}(\mathbf{m}^{(t)}, \mathbf{v}^{(t)}, \gamma^{(t)})$.

Corollary 4.8. A fixed point of the iterates of PAR-VARD satisfies the necessary KKT conditions in (4.4)–(4.6) and also (4.1)–(4.3).

Proof. Substituting a fixed point $m_j = m_j^{(t)}$ and $\gamma_k = \gamma_k^{(t)}$ into (4.7) and (4.1) reduces (4.7) to (4.4) and (4.1). Substituting a fixed point $v_j = v_j^{(t)}$ and $\gamma_k = \gamma_k^{(t)}$ into (4.8) and (4.2) reduces (4.8) to (4.5) and (4.2). Recall the definitions after (4.2) and (4.6). The condition in (4.6) is obtained by substituting $\mathbf{m}^{(t+1)} = \mathbf{m}$ and $\mathbf{v}^{(t+1)} = \mathbf{v}$ into (4.3).

The following theorem is due to Degirmenci and O’Sullivan.

Theorem 4.9. The objective function of (2.12)–(2.15) decreases monotonically during the iterations of PAR-VARD, and the decrease between subsequent iterations is lower bounded by

\( F^{(t)} - F^{(t+1)} \geq I\left( \frac{\beta^{(t)}}{Z^{(t)}} \parallel \frac{b^{(t)}}{Z^{(t)}} \right) + I\left( \frac{\tilde{\beta}^{(t)}}{Z_1} \parallel \frac{\tilde{b}^{(t)}}{Z_1} \right) + \sum_j g_j^{(t)} \left( \frac{1}{Z_j^{(t)}} \log \left( \frac{\tilde{z}_j^{(t)}}{\tilde{b}_j^{(t)}} \right) \right)^2 + \frac{1}{2} D_{IS}(\gamma^{(t)} || \gamma^{(t+1)}), \)

where $\beta^{(t)}$, $\tilde{\beta}^{(t)}$, and $Z^{(t)}$ are vectors with components $\beta_j^{(t)}$, $\tilde{\beta}_j^{(t)}$, and $Z_j^{(t)}$, defined in (4.10), (4.14) and (4.12), respectively. $b^{(t)}$, $\tilde{b}^{(t)}$ are vectors with components $b_j^{(t)}$ and $\tilde{b}_j^{(t)}$ defined in (3.5), and $g_j^{(t)}$ is defined in (3.14) ($g_j^{(t)} \geq 0$). Dividing two vectors should be interpreted as done elementwise.

Proof. The proof is very technical and therefore moved to Appendix A.

Several properties follow from Theorem 4.9 as stated in Theorem 4.10.

Theorem 4.10. Assume that $F^{(1)}$ is finite. Then

(a) The I-divergence $I\left( \frac{\beta^{(t)}}{Z^{(t)}} \parallel \frac{b^{(t)}}{Z^{(t)}} \right)$ converges to zero.

(b) The I-divergence $I\left( \frac{\tilde{\beta}^{(t)}}{Z_1} \parallel \frac{\tilde{b}^{(t)}}{Z_1} \right)$ converges to zero.

(c) The set of limit points of the mean iterates $\mathbf{m}^{(t)}$ is a connected set.

(d) The set of limit points of the variance iterates $\mathbf{v}^{(t)}$ is a connected set.

Proof. (a)-(b): From theorem 4.9 we have

\[
F^{(1)} - F^{(t)} = \sum_{s=1}^{t-1} [F^{(s)} - F^{(s+1)}] \geq \sum_{s=1}^{t-1} I\left( \frac{\beta^{(s)}}{Z^{(s)}} \parallel \frac{b^{(s)}}{Z^{(s)}} \right) + \sum_{s=1}^{t-1} I\left( \frac{\tilde{\beta}^{(s)}}{Z_1} \parallel \frac{\tilde{b}^{(s)}}{Z_1} \right) + \sum_{s=1}^{t-1} g_j^{(s)} \left( \frac{1}{Z_j^{(s)}} \log \left( \frac{\tilde{z}_j^{(s)}}{\tilde{b}_j^{(s)}} \right) \right)^2 + \sum_{s=1}^{t-1} \frac{1}{2} D_{IS}(\gamma^{(s)} || \gamma^{(s+1)}).
\]

As $t \to \infty$ all term on the right-hand side of (4.18) must go to zero since they are nonnegative and have a finite positive upper bound due to the fact that $F^{(1)}$ is finite and due to the monotonically decreasing and positive sequence $F^{(t)}$.

(c)-(d): Connectedness of the limit set of the iterates follows from convergence of the I-divergences to zero which implies by Pinsker’s inequality [38][Lemma 12.6.1] that $\parallel \beta^{(t)} -$
\( b(t) \|_1 \to 0 \) and \( \| \tilde{\beta}(t) - b(t) \|_1 \to 0 \). This together with (4.11) and (4.13) implies that \( m_j^{(t+1)} - m_j^{(t)} \to 0 \forall j \) and \( v_j^{(t+1)} - v_j^{(t)} \to 0 \forall j \).

We are now able to state the guarantees for global\(^7\) convergence of PAR-VARD.

**Theorem 4.11 (Global Convergence Theorem).** Let \( \{z(t)\}_{t=0}^{\infty} = \{(m(t), v(t), \gamma(t))\}_{t=0}^{\infty} \) be the sequence of iterates generated by Algorithm 1. Let the solution set \( \Gamma \) be the set of points \( (m, v, \gamma) \) which satisfy the KKT conditions in (4.4)–(4.6). Assume there exists \( z(1) \) such that \( F^{(1)} = F(z^{(1)}) \) is finite. Then

(a) The iterates are contained in a compact set.
(b) For \( z(t) \in \Gamma, F(t) \geq F^{(t+1)} \).
(c) For \( z(t) \notin \Gamma, F(t) > F^{(t+1)} \).
(d) The point-to-set mapping defined by Algorithm 1 is closed.
(e) All limit points of the iterates are in the solution set \( \Gamma \) and \( F(t) \to F(m, v, \gamma) \) as \( t \to \infty \) for some \( (m, v, \gamma) \in \Gamma \).

**Proof.** (a) By assumption, there is an initial guess \( z(1) \) such that \( F^{(1)} = F(z^{(1)}) \) is finite. From the \( \mathcal{F} \)-monotonicity of the sequence \( \{z(t)\}_{t=1}^{\infty} \), as asserted by theorem 4.9, and the positivity constraints incorporated into the algorithm, the iterates \( \{z(k)\}_{k=1}^{\infty} \) are contained in a sub-level set of \( \mathcal{F} \) given by \( \{ z : z \geq 0, F(z) \leq F^{(1)} \} \). To show that this set is compact we only need to show that \( \lim_{\|z\|_2 \to \infty} \mathcal{F}(z) = \infty \). Using (2.12)–(2.15), it can be verified that this condition is satisfied for any combination of \( m_j \to \infty, v_j \to \infty, \gamma_k \to \infty \).

(b) Follows from (4.17).

(c) Equivalently, we show that if \( F(t) \leq F^{(t+1)} \) then \( z(t) \in \Gamma \). Assume that \( F(t) \leq F^{(t+1)} \), then from (4.17) we must have \( F(t) = F^{(t+1)} \) and all \( I \) divergences in (4.17) must equal zero (as they are nonnegative). This implies that \( \beta_j^{(t)} \to b_j^{(t)} \) and \( \tilde{\beta}_j^{(t)} \to \tilde{b}_j^{(t)}, \forall j \). From (4.11) and (4.13) it follows that \( m_j^{(t+1)} \to m_j^{(t)} \), and \( v_j^{(t+1)} \to v_j^{(t)} \) so \( z(t) \) is a fixed point. It then follows from corollary 4.8 that \( z(t) \) satisfies the KKT conditions, i.e., \( z(t) \in \Gamma \).

(d) We use a result by Gunawardana and Byrne \[28\] which is restated here for convenience.

**Proposition 4.12 (Proposition 7 in \[28\]).** Given a real-valued continuous function \( f \) on \( A \times B \), define the point-to-set map \( F : A \to B \) by

\[
(4.19) \quad F(a) = \arg \min_{b \in B} f(a, b) = \{ b : f(a, b) \leq f(a, b') \quad \forall b' \in B \}.
\]

Then, the point-to-set mapping \( a \to F(a) \) is closed at \( a = a' \) if \( F(a') \) is nonempty.

The surrogate functions in (3.17)–(3.18) are continuous and convex, and they also satisfy \( S_j^m \geq 0 \) and \( S_j^v \geq 0 \) for any \( m_j, v_j \geq 0 \), thus insuring the existence of the solutions to (3.19)–(3.20). From Proposition 4.12 it follows that the mapping \( (m(t), v(t)) \to (m(t+1), v(t+1)) \) defined by (3.19)–(3.20) (lines 11-12 of Algorithm 1) is a closed point-to-set mapping. Performing several iterations of (3.19)–(3.20) also constitutes a closed mapping, since it is a composition of closed mappings. The mapping \( \gamma(t) \to \gamma(t+1) \) defined by (2.16) (line 15 of Algorithm 1) is a continuous and therefore closed point-to-point mapping. Finally, the composition of both

\(^7\)An algorithm is said to be “globally convergent” if for arbitrary starting points the algorithm is guaranteed to generate a sequence of points converging to a solution \[26\]. This does not imply a globally optimal solution.
closed mappings \((\mathbf{m}^{(t)}, \mathbf{v}^{(t)}, \gamma^{(t)}) \to (\mathbf{m}^{(t+1)}, \mathbf{v}^{(t+1)}, \gamma^{(t+1)})\) is also closed.

(e) Follows from (a)–(d) and Zangwill’s generalized convergence theorem \[35\].

5. Different Perspectives on VARD and Connections to Prior Art.

5.1. A Different View of VARD. We present an alternative view of the proposed VARD framework, which will provide further insight into the sparsity-promoting mechanism and will later enable extensions of the AM algorithm. The minimum of the objective function in (2.13)–(2.15) with respect to \(\gamma\) is given by

\[
\gamma_k^* = (\sum_j \psi_{kj} m_j)^2 + \sum_j \psi_{kj}^2 v_j.
\]

Substituting (5.1) into (2.13)–(2.15) we obtain an alternative formulation for VARD

\[
\begin{align*}
\min_{m \succeq 0, v \succeq 0} & \mathbb{E}_{q(x|m,v)} \left[ -\log p(y|x) \right] + \frac{1}{2} \sum_k \log(\mu_k^2 + \sigma_k^2) - h[q(x; m, v)], \\
\mu_k &= \sum_j \psi_{kj} m_j, \quad \sigma_k^2 = \sum_j \psi_{kj}^2 v_j, \quad h[q(x; m, v)] = \frac{1}{2} \sum_j \log v_j,
\end{align*}
\]

where the explicit expression for \(\mathbb{E}_{q(x|m,v)} \left[ -\log p(y|x) \right]\) is given by (2.13), and \(h[q(x; m, v)]\) in (5.3) denotes the differential entropy of the distribution \(q\). \(\mu\) and \(\sigma\) in (5.3) are the posterior mean and standard deviation (std) in the transform domain \(s = \Psi m\), respectively.

Note in (5.2) that the term \(\sum_k \log(\mu_k^2 + \sigma_k^2)\) promotes sparsity of both the posterior mean and standard deviation in the transform domain. In fact, without the entropy term \(h[q] = \sum_j \log v_j/2\) in (5.2), any \((\mu, \sigma)\) with at least one pair of components \((\mu_k, \sigma_k) = 0\), would be a global minimum of (5.2), with \(2^p\) possible combinations. The entropy term serves as a barrier preventing solutions with \(v_j = 0\), thus avoiding these unwanted minima. Note that the concave term \(\sum_k \log(\mu_k^2 + \sigma_k^2)\) in (5.2) is not separable with respect to the components of \(m\) and \(v\), and the convex decomposition approach used in Sec. 3 is not applicable here.

5.2. Connections to Reweighted \(\ell_2\) Algorithms. The ARD approach is related to the reweighted \(\ell_2\) algorithm [36] and the comparison between the two for Gaussian likelihoods has been presented in [37]. It is natural to consider the reweighted \(\ell_2\) algorithm for the Poisson likelihood as well, which consists of the following two steps

\[
\begin{align*}
x^{(t+1)} &= \arg \min_{x \geq 0} -\log p(y|x) + \frac{1}{2} \sum_k (\Psi x)_k^2 / \gamma_k^{(t)} \\
\gamma_k^{(t+1)} &= (\Psi x^{(t+1)})_k^2 / \gamma_k^{(t)} + \epsilon,
\end{align*}
\]

where \(\epsilon > 0\) is a fixed parameter that needs to be tuned. Note that we cannot choose \(\epsilon = 0\) since some entries in \(\Psi x\) are zero and this will result in infinite weights for the penalty in (5.4). For an easy comparison of (5.4)–(5.5) to the VARD algorithm in (2.10)–(2.11), we provide here again the update equations for VARD with the posterior variance fixed

\[
\begin{align*}
\mathbf{m}^{(t+1)} &= \min_{m \succeq 0} \mathbb{E}_{q(x|m,v)} \left[ -\log p(y|x) \right] + \frac{1}{2} \sum_k (\Psi m)_k^2 / \gamma_k^{(t)} \\
\gamma_k^{(t+1)} &= (\Psi m^{(t+1)})_k^2 + (\Psi \text{Diag}[\mathbf{v}^{(t+1)}] \Psi^T)_k.
\end{align*}
\]
It can be seen that in (5.4) the data-fidelity term is the negated log-likelihood, whereas for VARD in (5.6) it is the expectation of this term with respect to the approximate posterior \( q \). A key difference between the two algorithms is that in (5.5) a tuning parameter \( \epsilon \) is required, whereas in (5.7) the variances serve as “tuning parameters” with one parameter for each pixel/voxel, and are automatically learned from the data during the B-step.

Next, we make another type of comparison between the two approaches. We rewrite the reweighted \( \ell_2 \) algorithm in (5.4)–(5.5) as

\[
    x^{(t+1)} = \min_{x \geq 0} Q(x, \gamma^{(t)}) \quad \text{and} \quad \gamma^{(t+1)} = \min_{\gamma} \frac{Q(x^{(t+1)}, \gamma)}{\gamma},
\]

where \( Q \) is defined as

\[
    Q(x, \gamma) = -\log p(y|x) + \frac{1}{2} \sum_k [(\Psi x)_k^2 + \epsilon]/\gamma_k + \frac{1}{2} \sum_k \log \gamma_k.
\]

Minimizing (5.8) with respect to \( \gamma \) by solving \( \nabla_\gamma Q = 0 \) results in (5.5) with the superscript \((t + 1)\) removed. Substituting this solution into (5.8), we obtain the equivalent Type-I problem

\[
    \min_{x \geq 0} -\log p(y|x) + \frac{1}{2} \sum_k \log[(\Psi x)_k^2 + \epsilon].
\]

The result in (5.9) can be compared to the alternative formulation for VARD given in (5.2)–(3.3). Note that small values of \( \epsilon \) in (5.9) introduce many local minima, since any solution with at least one component \( (\Psi x)_k = 0 \) will result in a large negative penalty. In this case, any standard gradient-based algorithm will be easily trapped at a sub-optimal solution. On the other hand, large values of \( \epsilon \) avoid local minima but do not promote sparse solutions. The correct choice for the parameter \( \epsilon \) is therefore critical.

To facilitate the comparison between reweighted \( \ell_2 \) and PAR-VARD, we propose to modify the former using the separable surrogate approach as in Sec. 3, and derive the corresponding surrogate function. The surrogate function for the log-likelihood is given by

\[
    -\log p(y|x) + \frac{1}{2} \sum_j S_{j}^{\text{mle}}(x_j; x_j^{(t)}) \triangleq \sum_j \left\{ b_j^y x_j + \frac{b_j^{(t)}}{Z_1} \exp \left[ -Z_{\text{mle}}(x_j - x_j^{(t)}) \right] \right\},
\]

where \( b_j^{(t)} \) and \( b_j^y \) are computed according to (3.5) and (3.4) with \( \tilde{p}_j^{(t)} = 0 \) plugged in, and

\[
    Z_{\text{mle}} = \max_{i} \sum_j \phi_{ij}.
\]

\( S_{j}^{\text{mle}}(x_j) \) in (5.10) was derived in [11] and can be obtained as a special case of the PAR-VARD surrogate in (3.17) when \( m \) is replaced by \( x \), \( v \to 0 \), and \( \gamma \to \infty \), with consequently \( f_j^{(t)}, g_j^{(t)} \to 0 \). Adding the surrogate function for the quadratic penalty in (5.4), we obtain the update equation

\[
    x_j^{(t+1)} = \arg \min_{x \geq 0} S_j^x(x; x_j^{(t)}),
\]

\[
    S_j^x(x; x_j^{(t)}) = S_{j}^{\text{mle}}(x; x_j^{(t)}) + f_j^{(t)}(x - x_j^{(t)}) + g_j^{(t)}(x - x_j^{(t)})^2,
\]
We compare VARD to the AM algorithm for Type-I maximum likelihood estimation (MLE) in [11]. The model in [11] is beyond the scope of our work, but it can be reduced to the special case of a monoenergetic model with no background events given in (1.1). The surrogate function for MLE is given by (5.10). The minimizer of this surrogate is defined in (3.13)–(3.14) with \( m_j^{(t)} \) replaced by \( x_j^{(t)} \). The reweighted \( \ell_2 \) algorithm is then described in Algorithm 2. To the best of our knowledge, this algorithm has not been previously reported in the literature and can be considered as another contribution of this paper.

**Algorithm 2**: Reweighted \( \ell_2 \) with Separable Surrogates

1. Initialize \( x^{(1)} \), \( \gamma^{(1)} \)
2. Compute \( b^p \) in (3.5) % backprojection of data %
3. for \( t=1 \) to \( N \) do % AM iterations %
   4. \( p^{(t)} \leftarrow \Phi x^{(t)} \) % projections %
   5. \( \mu^{(t)} \leftarrow \eta \exp(-p^{(t)}) \)
   6. Compute \( g^{(t)} \), and \( f^{(t)} \) % defined in (3.13)–(3.14) with \( m_j^{(t)} \) replaced by \( x_j^{(t)} \) %
   7. \( b^{(t)} \leftarrow \Phi^T \mu^{(t)} \) % backprojections of \( \mu^{(t)} \) %
   8. for \( j=1 \) to \( p \) do % executed in parallel %
      9. \( x_j^{(t+1)} \leftarrow \arg \min_{x \geq 0} S^x(x; b_j^{(t)}, f_j^{(t)}, g_j^{(t)}) \) % see (5.12)–(5.13) %
   10. \( \gamma^{(t+1)} \leftarrow [\Psi x^{(t)}] \odot [\Psi x^{(t)}] + 1 \epsilon \) % \( 1 \) is the vector of all ones %

**Algorithm 3**: MLE/MAP with Separable Surrogates

1. Initialize \( x^{(1)} \)
2. Compute \( b^p \) in (3.5) % backprojection of data %
3. for \( t=1 \) to \( L \) do % convex decomposition iterations %
   4. \( p^{(t)} \leftarrow \Phi x^{(t)} \) % projections %
   5. \( \mu^{(t)} \leftarrow \eta \exp(-p^{(t)}) \) % predicted mean for measurements %
   6. \( b^{(t)} \leftarrow \Phi^T \mu^{(t)} \) % backprojections of \( \mu^{(t)} \) %
   7. for \( j=1 \) to \( p \) do % executed in parallel %
      8. \( x_j^{(t+1)} = \left[ x_j^{(t)} + \log(b_j^{(t)}/b_j^p)/Z_{\text{mle}} \right]_+ \) % this step is only for MLE %
      9. or \( x_j^{(t+1)} \leftarrow \arg \min_{x \geq 0} S_{\text{map}}(x; b_j^{(t)}) \) % this step is only for MAP (see (5.17)) %

where \( f_j^{(t)}, g_j^{(t)} \) are defined in (3.13)–(3.14) with \( \gamma_j^{(t)} \) given by (5.5) and \( m_j^{(t)} \) replaced by \( x_j^{(t)} \). The reweighted \( \ell_2 \) algorithm is then described in Algorithm 2. To the best of our knowledge, this algorithm has not been previously reported in the literature and can be considered as another contribution of this paper.

### 5.3. Comparison to Maximum Likelihood Estimation (MLE) and Maximum a Posteriori (MAP) Estimation.

We compare VARD to the AM algorithm for Type-I maximum likelihood estimation (MLE) in [11]. The model in [11] is beyond the scope of our work, but it can be reduced to the special case of a monoenergetic model with no background events given in (1.1). The surrogate function for MLE is given by (5.10). The minimizer of this surrogate is given in closed form, leading to the update equation

\[
\gamma_j^{(t+1)} = \left[ x_j^{(t)} + \frac{1}{Z_{\text{mle}}} \log \left( \frac{b_j^{(t)}}{b_j^p} \right) \right]_+,
\]

with \( b_j^{(t)} \) and \( b_j^p \) computed according to (3.5) and (3.4) with \( \tilde{p}_j^{(t)} = 0 \) plugged in, and \( Z_{\text{mle}} \) is defined in (5.11). The resulting algorithm is described in Algorithm 3.

It is also possible to add a penalty to the log-likelihood to obtain a MAP estimate [11]. In the following comparisons, we shall use the Huber-class neighborhood penalty given by
(1.4)–(1.5). The surrogate function for the Huber penalty involving the jth pixel is given by

\[ S_j^H(x; x_j^{(t)}) = \frac{\beta \delta^2}{N_j} \sum_{k: \{j,k\} \in \Theta} |x_j^{(t)} - x_k^{(t)} + N_j(x - x_j^{(t)})|/\delta - \log(|x_j^{(t)} - x_k^{(t)} + N_j(x - x_j^{(t)})|/\delta) \]

where \( N_j \) is the number of neighbors of the jth pixel. The updates for the MAP estimate are then given by

\[ x_j^{(t+1)} = \arg \min_{x \geq 0} S_j^{\text{map}}(x; x_j^{(t)}), \]

\[ S_j^{\text{map}}(x; x_j^{(t)}) = S_j^{\text{mle}}(x; x_j^{(t)}) + S_j^H(x; x_j^{(t)}), \]

where \( S_j^{\text{mle}} \) and \( S_j^H \) are given by (5.10) and (5.15), respectively. See Algorithm 3.

5.4. Connections to Previous Variational Inference Methods. The approximation of the posterior using a Gaussian distribution has been studied by Challis and Barber [39]. Although they did not consider the Poisson likelihood in (1.1) specifically, their approach is the same as our B-step in (2.10) if the posterior is chosen to be factorized and if \( \gamma \) are fixed and known. We emphasize that this is very different from the AM algorithm proposed here, which alternates between the B-step and the F-step where \( \gamma \) are also updated. In fact, a single B-step does not result in sparse solutions. In order to obtain sparse solutions, the B and F steps need to be repeated. This is similar in principle to the original EM-ARD algorithm [13], where a single E-step would not result in sparse solutions. Also, the parallel algorithm we propose in Sec. 3 is completely new. An alternative approach to [39] was proposed by Khan et al.[40] but it requires the inversion of a \( p \times p \) matrix (\( p \) is the number of pixels/voxels) so it is not feasible for large scale problems. We note that the method by Seeger and Nickisch [22] cannot be applied here for approximating the posterior since it is restricted to super-Gaussian likelihoods and the Poisson likelihood we consider (see (1.1)) is not super-Gaussian.

6. Generalization of VARD to Nonsquare \( \Psi \). Building on Sec. 5.1, we extend VARD to cases where the underlying sparse representation for the object is overcomplete, i.e., when \( \Psi \) in (2.1) has more rows than columns (previously we assumed \( \Psi \) to be square). Introducing a non-square \( \Psi \) in the AM algorithm of (2.10)–(2.11) results in a considerable complication of the \( \gamma \) update (F-step). The objective function includes the term \( -\log |\Psi^T \Gamma^{-1} \Psi| \) which can no longer be reduced to \( \sum_k \log \gamma_k \) as in (2.15) (recall that \( \Gamma = \text{Diag}(\gamma) \)). Taking the gradient of this expression with respect to \( \gamma \) yields \( \gamma^{-2} \circ \text{Diag}[\Psi^T (\Psi^T \Gamma^{-1} \Psi)^{-1} \Psi^T] \). Both the objective function and the gradient now have a computational complexity of \( O(p^3) \) (\( p \) is the number of pixels/voxels) which is prohibitive for large-scale problems.

To avoid the above complication, we use a different approach, where we start from the equivalent formulation in (5.2)–(5.3), and observe that \( \Psi \) can also be rectangular. Going back to the AM algorithm formulation, we obtain the same objective function as in (2.12)–(2.15), but with a rectangular \( \Psi \). However, the objective function can no longer be interpreted as the FVE in (2.5) for the prior in (2.1). This is due to the fact that we replaced the term \( -\log |\Psi^T \Gamma^{-1} \Psi| \) with \( \sum_k \log \gamma_k \). Since the difference between these two terms is independent of \( m \) and \( v \), minimizing the objective in (2.12)–(2.15) with respect to \((m, v)\) is still equivalent
to minimizing the KL divergence between the true posterior and the proposed posterior $q$. The F-step can no longer be interpreted as minimizing the KL divergence between $q$ and the prior, as in (2.11). Nevertheless, the promotion of sparsity is still asserted by the equivalent formulation in (5.2)–(5.3), where the $\sum_k \log(\mu_k^2 + \sigma_k^2)$ term promotes sparsity in the transform domain, and it is also supported by numerical experiments, provided in Sec. 7. It is important to note that although the interpretation of the objective function (as a function of $\gamma$) is different, it is given by the same expressions in (2.12)–(2.15), so the convergence analysis in Sec. 4 still applies.

To illustrate the type of priors that the above extension allows, we give an example for 2D-images where $\Psi$ is given by

$$\Psi = [\Psi^h; \Psi^v], \quad (\Psi^{h,v})_{jk} = \begin{cases} 1 & \text{if } j = k \\ -1 & \text{if } \{j, k\} \in \Theta^{h,v} \\ 0 & \text{otherwise} \end{cases},$$

where $[;]$ denotes concatenation, and $\Theta^h, \Theta^v$ define pairs of neighboring pixels located along the same horizontal or vertical lines in the image, with corresponding matrices $\Psi^h, \Psi^v$, respectively. We assign the same weight to horizontal and vertical pixel-differences, so $\gamma$ is replaced by $[\gamma; \gamma]$. It is also possible to incorporate differences between additional neighbors by concatenating additional $\Psi$ matrices for these additional neighbors in (6.1). It is also easily extendable to 3D images. To understand the difference between the above construction and the original one in (2.3), we write the sparsifying penalty from the objective function in (5.2) for both cases (considering only 2 neighbors per pixel for this example)

$$P_1 = \sum_{j=1}^{p} \log \left\{ \left( m_j - \frac{1}{2}(m_{j_1} + m_{j_2}) \right)^2 + v_j + \frac{1}{4}v_{j_1} + \frac{1}{4}v_{j_2} \right\},$$

$$P_2 = \sum_{j=1}^{p} \left[ \log \left\{ (m_j - m_{j_1})^2 + (v_j + v_{j_1}) \right\} + \log \left\{ (m_j - m_{j_2})^2 + (v_j + v_{j_2}) \right\} \right],$$

where $P_1$ corresponds to the original construction in (2.3) and $P_2$ corresponds to the construction in (6.1); $j_1$ and $j_2$ denote the indexes of the horizontal and vertical neighbors of the $j$th pixel, respectively, i.e., $\{j, j_1\} \in \Theta^h$ and $\{j, j_2\} \in \Theta^v$. For small variances, $P_2$ promotes small differences along the horizontal and vertical axes separately (similarly to (1.4)). In contrast, $P_1$ promotes small differences between a pixel and the average of its neighbors.

7. Numerical Results. In the following section we study the performance of the proposed PAR-VARD algorithm for x-ray CT. For medical applications, there is a limit on the intensity of the x-ray source to avoid harmful radiation levels. We consider both simulated and real data for x-ray CT with source intensity levels ($\eta$ values in (1.1)) which are typical of clinical settings. The geometry considered in the experiments is shown in Fig. 4.

We compare the following methods: (1) The proposed PAR-VARD algorithm described in algorithm 1; (2) Maximum likelihood estimator (MLE) described in algorithm 3; (3) Maximum a Posteriori (MAP) estimator with the Huber neighborhood penalty described in algorithm 3; (4) The reweighted $\ell_2$ algorithm described in Algorithm 2. For simplicity, we shall abuse
notation slightly and refer to PAR-VARD as VARD. We compare two different priors corresponding to two choices of $\Psi$ in (2.3) and (6.1), which we refer to as “complete” (C) and “over-complete” (O) representations, respectively. All algorithms were implemented in MATLAB and executed on a desktop PC with 4 Intel Xeon E7 CPU’s (8 cores each) running a Linux platform. In our implementation, the line searches for each pixel are done in parallel using 32 cores. All algorithms were run till convergence with the objectives being flat.

7.1. Simulated Data. The simulated data were generated for a Shepp-Logan phantom with Poisson noise according to the model in (1.1) and source intensities $\eta = 10^4, 10^5$, which are typical of clinical systems. We use an image resolution of $256 \times 256$, which for the geometry specified in Fig. 4 required 1372 view-angles (full scan around the object) and 512 detectors for each view (702,464 measurements in total). The number of measurements were calculated according to classical sampling theory as detailed in [2].

The root-mean-squared errors (RMSE) obtained with the different methods is shown in Tables 1–2, with source intensity set to $\eta = 10^5$ and $\eta = 10^4$, respectively. For MAP we had to choose values for two different tuning parameters in (1.5), and perform many trials. For reweighted $\ell_2$ we had to choose values for one tuning parameter in (5.5) and run multiple trials as well. One can see that VARD with the over-complete representation (VARD-O) yields RMSE that is comparable to the best result with the alternative methods but without any tuning, and using only a single reconstruction. By comparing Tables 1(a),(c) and Tables 2(a),(c), one observes that both reweighted $\ell_2$ and MAP are significantly more sensitive to the choice of tuning parameters when the noise level is higher. Both methods result in much higher RMSE than VARD when the tuning parameters are not properly chosen.

It is important to note that in practice the object is unknown (RMSE cannot be computed) so the tuning parameters are chosen by examining the reconstructed image for each and every trial, which is very time consuming. The purpose of the comparison in Tables 1–2 is merely to show the sensitivity of the MAP and reweighted $\ell_2$ estimators to the particular choice of the tuning parameters. The lowest RMSE is obtained using reweighted $\ell_2$ but the differences in the image compared to VARD are visually very minor, as shown below.

Some of the reconstructed images using these methods are shown Fig. 5. For MAP and reweighted $\ell_2$ we present the results for two different choices of tuning parameters, one of which is the best result we could find. One can see in Figs. 5(d) and 5(f) how significant
artifacts occur when the tuning parameters are not chosen correctly. In contrast, the VARD reconstruction in Fig. 5(g) does not have any significant artifacts and the image quality is comparable to the best results obtained with the alternative methods (Figs. 5(c), 5(e)). The reconstructed hyperparameters $\gamma$ are shown in Fig. 6(a). One observes that $\gamma$ has large values at discontinuities and very low values in regions where the attenuation levels are constant. Recall that VARD with the prior in (6.1) promotes sparsity in the pixel-difference domain. Figure 6(a) clearly indicates that VARD has learned the correct support of the object in the pixel-differences domain. The posterior std is shown in Fig. 6(b) and has a similar structure to $\gamma$. Note that one should be careful when interpreting the posterior std since the support of the posterior is $(-\infty, \infty)$ (see (2.9)) and the std can extend to negative attenuation values.

| (a) MAP | (b) MLE/VARD |
|---------|--------------|
| $\beta = 10^5$ | 1.79 \% | 1.52 \% | 0.65 \% | 2.10 \% |
| $\beta = 10^6$ | 1.51 \% | 0.54 \% | 1.16 \% | 9.26 \% |

| (c) Reweighted $\ell_2$ |
|------------------------|
| $\epsilon = 10^{-8}$ | 11.03 \% | 0.66 \% | 1.32 \% | 1.81 \% |
| $\epsilon = 10^{-6}$ | 2.88 \% | 0.41 \% | 0.99 \% | 1.79 \% |

| (a) RMSE using the MAP estimator with a Huber penalty for different values of tuning parameters; (b) RMSE for the MLE estimator and VARD; (c) RMSE for reweighted $\ell_2$ with different $\epsilon$ values. C-“Complete” and O-“over-complete” indicate the use of $\Psi$ in (2.3) and (6.1), respectively. Here $\eta = 10^5$. MAP, Reweighted $\ell_2$, and VARD use 1 horizontal and 1 vertical neighbor per pixel. All algorithms were run till 2000 iterations to insure convergence. The standard deviation (std) of the RMSE over 50 different noise realizations was negligible and therefore was not taken into account, e.g., for VARD-O std was $\approx 0.015\%$.

| (a) MAP | (b) MLE/VARD |
|---------|--------------|
| $\beta = 10^5$ | 5.29 \% | 3.01 \% | 1.51 \% | 9.30 \% |
| $\beta = 10^6$ | 2.99 \% | 1.36 \% | 7.27 \% | 40.00 \% |

| (c) Reweighted $\ell_2$ |
|------------------------|
| $\epsilon = 10^{-8}$ | 46.2 \% | 3.54 \% | 2.49 \% | 5.23 \% |
| $\epsilon = 10^{-6}$ | 21.34 \% | 1.5 \% | 1.16 \% | 4.80 \% |
Figure 5. Reconstructions of attenuation images (relative to water) from synthetic data with Poisson noise using different methods. All images have resolution $256 \times 256$. (a) Truth; (b) Maximum likelihood (MLE) estimate; (c) Maximum a posteriori (MAP) estimate using the Huber-class neighborhood penalty with $\beta = 10^6$ and $\delta = 10^{-4}$; (d) same as Fig. (c) but with $\beta = 10^7$ and $\delta = 10^{-3}$; (e) Reweighted $\ell_2$-O with $\epsilon = 10^{-6}$; (f) same as (e) but with $\epsilon = 10^{-8}$; (g) VARD-O; (h) objective functions for all methods. All images are displayed using the window $[0.15, 0.35]$, even though the phantom includes values in $[0, 1]$, in order to enhance the differences between the different methods. A zoom-in view of the region inside the small red frame is shown in the inset to the left in figures (a)–(g). All algorithms except MLE use a neighborhood penalty with one horizontal and one vertical neighbor per pixel. All algorithms were run till 2000 iterations to insure convergence. We recommend viewing the figures with at least 150% zoom-in.
In Sec. 7.2 we show a useful interpretation of the posterior variances.

All algorithms were run till 2000 iterations to insure convergence. The images were initialized by setting all pixel values to zero. For VARD, all posterior variances \( v \) were initialized to 1. For VARD and reweighted \( \ell_2 \), the hyperparameters \( \gamma \) were initialized to 100 to make the data-fit term dominant in the first iteration. Figure 5(h) presents the objective function for VARD-O ((2.12)–(2.15)) vs. number of iterations, as well as the objectives for the other methods. The objective is seen to decrease monotonically, as predicted by theory, and can be used to assess convergence in practice. All methods share similar convergence rates.

![Figure 6. Reconstructed standard deviations (std) using VARD on synthetic data. (a) squared root of hyperparameters \( \gamma \); (b) squared root of posterior variances \( v \).](image)

7.2. Real Data. Next we present reconstructions for a dataset acquired at an experimental laboratory at Duke University with the geometry specified in Fig. 4. The number of views was 360 (full scan around the object) with view-angle resolution of 1°. For each view 1781 detector pixels were used. The total number of measurements was 641,160. Additional details about the experimental setup are provided in Appendix C. We imaged a 2D cross section of an acrylic glass cylinder full of cylindrical holes with different diameters. In this case, the image for the ground truth is unknown. However, a good estimate of the truth is based on the specifications from the manufacturer of this phantom. The attenuation of the acrylic glass relative to water is 1.5, and for the inside of the rods it is 0 (air).

Figure 7 presents reconstructions using different methods. Image resolution is 256 × 256. For MAP and reweighted \( \ell_2 \) we show two examples of reconstructions, one for the best choice of the tuning parameter that we could find, and one with significant artifacts for a bad choice of tuning parameters. Again, one can see how VARD produces images which are comparable in quality to the best trial with MAP or reweighted \( \ell_2 \) but without any need for repeated trials. Figure 8 shows the reconstructed variances, where one can see again how VARD learns the correct support of the object in the pixel-differences domain.

Finally, in Figs. 9–10 we present an example which illustrates the robustness of VARD to missing data and also demonstrates a useful interpretation of the posterior variances. In this example we scanned a “DUKE” letters object suspended in mid-air using the same experimental system. The letters are made from VeroBlue material with a linear attenuation
Figure 7. Reconstructions of attenuation maps (relative to water) from real data using different methods. (a) Maximum likelihood (MLE) estimate; (b) VARD-O; (c) Maximum a posteriori (MAP) estimate using the Huber-type neighborhood penalty with $\beta = 10^3$ and $\delta = 0.1$; (d) Same as (c) but with $\beta = 10^2$; (e) Reweighted $\ell_2$-O with $\epsilon = 10^{-5}$; (f) Same as (e) but with $\epsilon = 10^{-3}$. All algorithms use 2000 iterations to insure convergence. All except MLE use a neighborhood penalty with one horizontal and one vertical neighbor per pixel. We recommend viewing the figures with at least 150% zoom-in. All objectives were monotonically decreasing and flat when the algorithms were terminated.
coefficient of 1.458 relative to water (more details in caption of Fig. 9). The expected image is sparse so we used the prior in (2.1) with $\Psi = I$ to promote sparsity in the pixel basis.

We subsampled the angular grid for the source-locations (views) so only 1/8 or 1/16 of the views are used; the VARD reconstructions for these two cases are shown in Fig. 9(a) and Fig. 9(b), respectively. In addition, we include in Fig. 9(c) and 9(d) the reconstructions using filtered-back-projection (FBP) [1], which is a linear reconstruction method based on Fourier inversion. Note that the FBP reconstruction exhibits much more prominent artifacts than VARD in the form of streaks, which are due to view undersampling. We present the FBP reconstructions again in Figs 10(a)–10(b) where the displayed attenuation values were limited to a smaller range in order to enhance the artifacts. There is a striking resemblance between the streaks in the posterior std images obtained by VARD in Figs 10(c)–10(d) and the streaks in the FBP reconstructions in Figs 10(a)–10(b). This demonstrates that the variances can indicate regions in the image which are insufficiently sampled. A more detailed study of this relationship will be deferred to a subsequent publication.

Lastly, we note that although we have focused on the 2D fan-beam geometry in the numerical experiments, the proposed method can be applied to any other geometry.

7.3. Implementation Details. In our implementation, the line searches for each pixel (lines 11–12 in Algorithm 1, line 9 in Algorithm 2 and Algorithm 3) are done in parallel and using Newton’s method. Since it is not necessary to solve the 1D surrogate problems (line searches) exactly, we found that it is more efficient to perform just one Newton step for each iteration. However, for MAP this leads to divergence whenever the solution is in the linear regime of the Huber penalty. Therefore, we use a 1D trust-region Newton method (see details in Appendix B), which takes more time to execute than a Newton step. For VARD we could use Newton steps for the mean updates, but had to resort to the trust-region method for the variance updates (see details in Appendix B) to insure decrease of the objective. For reweighted $\ell_2$ the Newton steps worked successfully. In our implementation, the run time was dominated by the trust-region updates, so both VARD and a single trial of MAP took the same time to converge, which was 40 minutes for the synthetic dataset (35 min for the acrylic
Figure 9. Reconstructions from real data. (a) VARD with sparsity promoted in the pixel basis and using 1/8 of the views; (b) same as (a) but using 1/16 of the views; (c) Filtered back projection using 1/8 of the views; (d) same as in (c) but using 1/16 of the views. For FBP, we used cubic interpolation for the backprojection, and a cropped Ram-Lak (ramp) filter. Image resolution is 128 × 128. The full dataset (before the view subsampling) was acquired using a full 360° scan around the object with view-angle resolution of 1° and 1573 detector pixels per view. 510 iterations were used for VARD.

glass phantom). A single trial of reweighted ℓ2 took 20 minutes to converge.

For the problem sizes considered here, we could store the system matrix and its transpose in memory using MATLAB’s compressed sparse representation. The time for the forward/back-projections was negligible relative to the trust-region Newton updates. For larger problems, it might not be possible to store the matrices in memory and an “on-the-fly” implementation of the forward/back-projection operations would be required and could take a larger portion of the run time than the line searches. This could lead to comparable run times for MAP and reweighted ℓ2 (as they both require the same number of forward/back-projections per iteration). Also, in cases where the computational resources are limited, VARD could be slower than a single MAP or reweighted ℓ2 trial up to a factor of 2, since VARD requires twice the number of forward/backprojections (compare Algorithm 1 with Algorithms 2–3). However, considering the large number of trials that needs to be done when using methods with tuning parameters, VARD still has a lower total run-time than the alternative methods.
8. Conclusions. We presented a new framework for automatic relevance determination (ARD), which we call variational ARD (VARD). It provides an extension of previous ARD methods to Poisson models used in transmission tomography. VARD computes approximations to the posterior mean and variance, with the former being the estimate for the object. We revealed the sparsity-promoting mechanism in VARD, and established key differences compared to previous ARD work that are applicable only to Gaussian likelihoods. An important advantage of the VARD framework is that it avoids the cumbersome process of choosing tuning parameters, while still retaining good image quality. We have shown that the posterior variances play a key role in learning the correct balance between the data-fit and the prior. Straightforward simplifications as reweighted $\ell_2$ do not retain this property.

We presented a parallel AM algorithm for VARD (PAR-VARD) which is applicable to large scale problems. We studied a parallel implementation of PAR-VARD using both synthetic and real data. The numerical experiments demonstrate that the images produced by PAR-VARD are comparable in quality to the best MAP and reweighted $\ell_2$ estimates. However, PAR-VARD requires much less time when taking into account the many trials required by
MAP and reweighted $\ell_2$ and the time spend on inspecting the images from different trials. As apposed to previous ARD methods, the exact evaluation of the objective for PAR-VARD is practically feasible for large scale problems and can be used to assess convergence in practice. We provided a detailed study of the convergence properties of PAR-VARD and established guarantees for global convergence.

In addition to the estimate of the object, VARD also computes posterior variances. We have shown that when the object is undersampled, the variances can indicate regions in the image where sampling is insufficient. The variances can therefore be used to define relative levels of confidence in the reconstruction at particular regions in the image. The variances can also be used for adaptive sensing/experimental design [22]. Our results suggest that the variances can be used for image segmentation or boundary detection. These potential uses are not studied here, and will be the subject of future work.

We have explored the use of ordered subsets (OS) [6, 11] with PAR-VARD, where the dataset is split into a number of subsets, each processed in a predefined order. This led to considerable speed-ups, similar to the ones reported in the literature for other algorithms using OS. To focus on the algorithm presented in this paper we chose not to present OS results here.

9. Acknowledgments. We would like to thank Prof. Martin P. Tornai for providing us access to the Duke Multi-Modality Imaging Laboratory (MMIL) where the experimental data was acquired. We would like to thank Mr. Andrew Holmgren for conducting the experiments. We thank Dr. Ikenna Odinaka for suggesting the 1D trust-region Newton method used in our code. We thank Mr. David Carlson for proofreading the manuscript and providing helpful comments. This work was supported by the Department of Homeland Security, Science and Technology Directorate, Explosives Division, through contract HSHQDC-11-C-0083.

Appendix A. Proof of Theorem 4.9.

Proof. We split the difference in (4.17) into two terms

$$F(m^{(t)}, v^{(t)}, \gamma^{(t)}) - F(m^{(t+1)}, v^{(t+1)}, \gamma^{(t+1)}) = F(m^{(t)}, v^{(t)}, \gamma^{(t)}) - F(m^{(t+1)}, v^{(t+1)}, \gamma^{(t+1)})$$

(A.1)

$$+ F(m^{(t+1)}, v^{(t+1)}, \gamma^{(t+1)}) - F(m^{(t+1)}, v^{(t+1)}, \gamma^{(t+1)}).$$

First we focus on the first difference in (A.1) and write it explicitly using (3.4)-(3.5)

$$F(m^{(t)}, v^{(t)}, \gamma^{(t)}) - F(m^{(t+1)}, v^{(t+1)}, \gamma^{(t+1)}) = \sum_j \beta_j^p (m_j^{(t)} - m_j^{(t+1)}) + \sum_i \mu_i^{(t)} -$$

$$\sum_i \mu_i^{(t)} \exp \left[ -\sum_j \phi_{ij}(m_j^{(t+1)} - m_j^{(t)}) + \sum_j \phi_{ij}^2 (v_j^{(t+1)} - v_j^{(t)})/2 \right] +$$

$$\sum_k \frac{1}{2\gamma_k} \left[ (\sum_j \psi_{kj}^{(t)})^2 - (\sum_j \psi_{kj} m_j^{(t+1)})^2 \right] +$$

$$\sum_k \frac{1}{2\gamma_k} \left[ \sum_j \psi_{kj}^2 (v_j^{(t+1)} - v_j^{(t)}) \right] + \sum_j \log (v_j^{(t+1)}/v_j^{(t)})/2.$$

(A.2)
Next, we present several bounds for terms in (A.2). Using (3.7)–(3.8) and (3.5) we obtain

\begin{equation}
\sum_i \mu_i^{(t)} \geq \sum_i \mu_i^{(t)} \sum_j \phi_{ij} + \phi_{ij}^2 / Z_1 = \sum_j b_j^{(t)} / Z_1 \geq \sum_j \tilde{b}_j^{(t)} / Z_1,
\end{equation}

where the last inequality is based on the definition for \( Z_j^{(t)} \) in (4.12). We bound the exponential term in (A.2) using the convex decomposition lemma (lemma 3.1)

\begin{equation}
- \sum_i \mu_i^{(t)} \exp \left( - \sum_j \phi_{ij} (m_j^{(t+1)} - m_j^{(t)}) + \sum_j \phi_{ij}^2 (v_j^{(t+1)} - v_j^{(t)}) / 2 \right) \geq - \sum_j b_j^{(t)} \exp \left( - Z_j^{(t)} (m_j^{(t+1)} - m_j^{(t)}) \right) \geq \sum_j \tilde{b}_j^{(t)} \exp \left( Z_1 (v_j^{(t+1)} - v_j^{(t)}) \right),
\end{equation}

where we used (3.8) with \( Z_1 \) replaced by \( Z_j^{(t)} \geq Z_1 \) in \( r_{ij} \) (lemma 3.1 with the dummy variable extension applied), and also used (3.5). Using lemma 3.1 for the quadratic term in (A.2) we obtain

\begin{equation}
\sum_k 1 / 2 \gamma_k \left[ (\sum_j \psi_k m_j^{(t+1)})^2 - (\sum_j \psi_k m_j^{(t+1)})^2 \right] \geq - \sum_j \left[ f_j^{(t)} (m_j^{(t+1)} - m_j^{(t)}) - g_j^{(t)} (m_j^{(t+1)} - m_j^{(t)})^2 \right],
\end{equation}

where the derivation is similar to the one in (3.10) and we also used (3.13)–(3.14). Substituting (4.11), (4.13) into the right hand sides of (A.4) and (A.5) and then substituting the result into (A.2), we obtain

\begin{equation}
\mathcal{F}(m^{(t)}, v^{(t)}, \gamma^{(t)}) - \mathcal{F}(m^{(t+1)}, v^{(t+1)}, \gamma^{(t)}) \geq \sum_j \left\{ b_j^{(t)} \log \left( \frac{b_j^{(t)}}{b_j^{(t)}} \right) + \frac{b_j^{(t)}}{Z_j^{(t)}} + \frac{\tilde{b}_j^{(t)}}{Z_1} - \frac{\beta_j^{(t)}}{Z_j^{(t)}} - \frac{\tilde{\beta}_j^{(t)}}{Z_1} \log \left[ \frac{\beta_j^{(t)}}{\tilde{\beta}_j^{(t)}} \right] + \log \left( \frac{\tilde{\beta}_j^{(t)}}{\beta_j^{(t)}} \right) \right\}.
\end{equation}

We also use

\begin{equation}
(m_j^{(t+1)} - m_j^{(t)}) = 2(\tilde{m}_j^{(t)} - m_j^{(t)}) + (m_j^{(t+1)} - m_j^{(t)} + 2(m_j^{(t)} - \tilde{m}_j^{(t)})) =
\end{equation}

\begin{equation}
2(\tilde{m}_j^{(t)} - m_j^{(t)}) + \frac{1}{Z_j^{(t)}} \log \left( \frac{\beta_j^{(t)}}{\tilde{\beta}_j^{(t)}} \right) \geq 2(\tilde{m}_j^{(t)} - m_j^{(t)}) + \frac{1}{Z_j^{(t)}} \log \left( \frac{\beta_j^{(t)}}{\tilde{\beta}_j^{(t)}} \right),
\end{equation}

where we have used (4.7) and (4.11). Next, we use (4.10) in (A.7) to obtain

\begin{equation}
(m_j^{(t+1)} - m_j^{(t)}) \geq (\beta_j^{(t)} - \tilde{\beta}_j^{(t)}) / g_j^{(t)} + \frac{1}{Z_j^{(t)}} \log \left( \frac{\beta_j^{(t)}}{\tilde{\beta}_j^{(t)}} \right).
\end{equation}
and substituting into (A.6) we have
\[
\mathcal{F}(m^{(t)}, v^{(t)}, \gamma^{(t)}) - \mathcal{F}(m^{(t+1)}, v^{(t+1)}, \gamma^{(t)}) \geq \frac{b_j^{(t)}}{Z_j^{(t)}} + \frac{\tilde{b}_j^{(t)}}{Z_j^{(t)}} - \frac{\tilde{\beta}_j^{(t)}}{Z_1} + \sum_j g_j^{(t)} \left[ \frac{1}{Z_j^{(t)}} \log \left( \frac{\beta_j^{(t)}}{\tilde{\beta}_j^{(t)}} \right) \right]^2 + \beta_j^{(t)} \log \left( \frac{\beta_j^{(t)}}{\tilde{\beta}_j^{(t)}} \right) - \frac{\tilde{\beta}_j^{(t)}}{Z_1} + \frac{1}{2} \log \left( \frac{v_j^{(t+1)}}{v_j^{(t)}} \right).
\]

(A.9)

Finally, we use \( \log(x) \geq 1 - 1/x \) applied to the last term in (A.9), which gives
\[
\log(v_j^{(t+1)}/v_j^{(t)}) \geq \frac{1}{v_j^{(t+1)}}(v_j^{(t+1)} - v_j^{(t)}) = \frac{1}{v_j^{(t+1)}} \frac{1}{Z_1} \log \left( \frac{\tilde{\beta}_j^{(t)}}{\tilde{\beta}_j^{(t)}} \right),
\]

where we used (4.13). Substituting (A.10) into (A.9), using (4.14) and the definition of the I-divergence in (4.15), we obtain
\[
\mathcal{F}(m^{(t)}, v^{(t)}, \gamma^{(t)}) - \mathcal{F}(m^{(t+1)}, v^{(t+1)}, \gamma^{(t)}) \geq \frac{1}{\gamma_k^{(t)}} \left[ \sum_j \psi_{kj} m_j^{(t+1)} \right] + \frac{1}{2} \sum_j \log \left( \frac{\gamma_k^{(t)}}{\gamma_k^{(t+1)}} \right).
\]

(A.11)

Next, we calculate the second difference in (A.1)
\[
\mathcal{F}(m^{(t+1)}, v^{(t+1)}, \gamma^{(t)}) - \mathcal{F}(m^{(t+1)}, v^{(t+1)}, \gamma^{(t+1)}) = \frac{1}{2} \sum_k \left( \frac{1}{\gamma_k^{(t+1)}} - \frac{1}{\gamma_k^{(t)}} \right) \left[ \sum_j \psi_{kj} m_j^{(t+1)} \right] + \frac{1}{2} \sum_j \log \left( \frac{\gamma_k^{(t)}}{\gamma_k^{(t+1)}} \right).
\]

(A.12)

Substituting (2.16) into (A.12) we obtain
\[
\mathcal{F}(m^{(t+1)}, v^{(t+1)}, \gamma^{(t)}) - \mathcal{F}(m^{(t+1)}, v^{(t+1)}, \gamma^{(t+1)}) = \frac{1}{2} \sum_k \log \left( \frac{\gamma_k^{(t+1)}}{\gamma_k^{(t)}} \right) + \frac{1}{2} \left( \frac{\gamma_k^{(t+1)}}{\gamma_k^{(t)}} - 1 \right) \overset{!}{=} \frac{1}{2} D_{IS}(\gamma^{(t)} || \gamma^{(t+1)}),
\]

where we used the definition in (4.16). Combining the results in (A.11) and (A.13) we have
\[
\mathcal{F}(m^{(t)}, v^{(t)}, \gamma^{(t)}) - \mathcal{F}(m^{(t+1)}, v^{(t+1)}, \gamma^{(t+1)}) = \mathcal{F}(m^{(t)}, v^{(t)}, \gamma^{(t)}) - \mathcal{F}(m^{(t+1)}, v^{(t+1)}, \gamma^{(t+1)}) + \mathcal{F}(m^{(t+1)}, v^{(t+1)}, \gamma^{(t+1)}) - \mathcal{F}(m^{(t+1)}, v^{(t+1)}, \gamma^{(t+1)}). \geq \mathcal{I} \left( \frac{\beta^{(t)}}{Z^{(t)}} \right) \left\| \frac{b^{(t)}}{Z^{(t)}} \right\| + \frac{1}{2} \mathcal{I} \left( \frac{\beta^{(t)}}{Z_1} \right) \left\| \frac{b^{(t)}}{Z_1} \right\| + 2 \sum_j g_j^{(t)} \left[ \frac{1}{Z_j^{(t)}} \log \left( \frac{\beta_j^{(t)}}{\tilde{\beta}_j^{(t)}} \right) \right]^2 + \frac{1}{2} D_{IS}(\gamma^{(t)} || \gamma^{(t+1)}),
\]

(A.14)

which is the result stated in Theorem 4.9. ■
Appendix B. Implementation Details. Next, we address the implementation of lines 11-12 in algorithm 1 which involve a line search to minimize the surrogate function. Since it is not necessary to find the exact solution for the surrogate, we use a single Newton step for the mean with thresholding, i.e.,

\[
(m^{(t+1)}_{j}) = \left[ m^{(t)}_{j} - \left( \frac{\partial_{x}S(x)}{\partial_{x}^{2}S(x)} \right)_{x=m^{(t)}_{j}} \right]_{+} = \left[ m^{(t)}_{j} - \left( b^{y}_{j} - b^{(t)}_{j} + f^{(t)}_{j} \right) / \left( Z_{1}b^{(t)}_{j} + 2g^{(t)}_{j} \right) \right]_{+},
\]

where \( \partial_{x} \) denotes the derivative with respect to \( x \), \( S \) denotes the surrogate in (3.17), and all parameters are defined in (3.5), (3.8),(3.13),(3.14). While Newton steps as in (B.1) are not guaranteed to decrease the objective function, we have found that in all our numerical experiments the objective function decreased at each iteration. If the objective function is not monotonically decreasing, then the step size can be adjusted or a trust-region approach can be used, as described below for the variance updates. For the variance updates, we use a Newton trust region approach \[41\] where Newton steps for \( v_{j} \) are considered (similarly to (B.1)) but the steps are limited to within a trusted region. At each iteration, the trust region is adjusted from both left and right by comparing predicted reduction based on a local quadratic approximation and the actual reduction. If the prediction is poor, the trust region is shrunk. The left boundary is limited to the positive axis. This approach also insures the decrease of the objective function at each step.

For MAP with Huber neighborhood penalties we found that a trust region approach is also required to guarantee convergence in cases where the linear part of the penalty dominates. For reweighted \( \ell_{2} \), taking Newton steps was sufficient.

Appendix C. Details for the Experimental Lab Setup. The real data has been acquired at the Duke Multi-Modality Imaging Lab (MMIL) \[42\]. The x-ray source specifications are 60kVp, 50mA, and 25ms. A 0.55mm Cerium filter was used to strongly filter the Bremsstrahlung (continuous spectrum x-rays) and create a pseudo monoenergetic source spectrum. The detector has a pixel pitch of 127 \( \mu \text{m} \) and a fill-factor of 0.75 (pixel pitch divided by pixel size). By integrating the flux-energy curve computed using the XSPECT software \[43\] and accounting for detector fill-factor, we estimated a total of \( \eta = 2 \times 10^{3} \) photons per detector pixel. The view angle resolution is 1\(^{\circ}\).

REFERENCES

[1] A. Kak and M. Slaney, Principles of Computerized Tomographic Imaging, SIAM, 1988.
[2] F. Natterer, The Mathematics of Computerized Tomography, John Wiley & Sons, 1986.
[3] W. A. Kalender, X-ray computed tomography, Phys. Med. Biol., 51(13), 2006, pp. R29-R43.
[4] J. Frank, Electron tomography, Springer, 1992.
[5] M. H. Li, H. Q. Yang, and H. Kudo, An accurate iterative reconstruction algorithm for sparse objects: Application to 3d blood vessel reconstruction from a limited number of projections, Phys. Med. Biol., 47, 2002, pp. 2599-2609.
[6] J. A. Fessler, Handbook of Medical Imaging Vol. 2, Chapter 1: Statistical Image Reconstruction Methods for Transmission Tomography, Edited by M. Sonka and J. M. Fitzpatrick, SPIE, 2000.
[7] E. P. Simoncelli, Modeling the joint statistics of images in the wavelet domain, Proc. SPIE 1999, pp. 188–195.
[8] L. Boubchir and J. M. Fadili, Multivariate statistical modeling of images with the curvelet transform, Proc. 8th International Conference on Signal Proc. and Its Applications, 2005, pp. 747–750.

[9] K. Lang and R. Carson, EM reconstruction algorithms for emission and transmission tomography, J. Comput. Assist. Tomog., 8(2), 1984, pp. 306–316.

[10] K. Lange and J. A. Fessler, Globally convergent algorithms for maximum a posteriori transmission tomography, IEEE Trans. Image Proc., 4(10), 1995, pp. 1430–1438.

[11] J. A. O’Sullivan and J. Benac, Alternating minimization algorithms for transmission tomography, IEEE Trans. Med. Imaging, 26(3), 2007, pp. 283-297.

[12] R. M. Neal, Bayesian Learning for Neural Networks, Springer-Verlag, 1996.

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

[14] D. P. Wipf, B. D. Rao, and S. Nagarajan, Latent variable Bayesian models for promoting sparsity, IEEE Trans. Inform. Theory, 57, 2011, pp. 6236–6255.

[15] W. Zhuang, S. S. Gopal, and T. J Hebert, Numerical evaluation of methods for computing tomographic projections, IEEE Trans. Nucl. Sci., 4, 1994, pp. 1660–1665.

[16] I. Csiszár and G. Tusnády, Information geometry and alternating minimization procedures, Statistics and Decisions, Supplementary Issue 1, 1984, pp. 205-237.

[17] Y. Sidky, C. M. Kao, and X. H. Pan, Accurate image reconstruction from few-views and limited-angle data in divergent-beam CT, J. X-ray Sci. Tech., 14, 2006, pp. 119–139.

[18] Y. Sidky and X. C. Pan, Image reconstruction in circular cone-beam computed tomography by constrained total-variation minimization, Phys. Med. Biol., 53, 2008, pp. 4777–4807.

[19] D. P. Wipf, J. Palmer, and B. Rao, Perspectives on sparse Bayesian learning, Proc. Neural Information Processing Systems, 16(1), 2004, pp. 249-256.

[20] D. P. Wipf and S. Nagarajan, A new view of automatic relevance determination, Proc. Neural Information Processing Systems, 20, 2008.

[21] A. P. Dempster, N. M. Laird, and D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. Roy. Stat. Soc., 39(1), 1977, pp. 1–38.

[22] M. Seeger and H. Nickisch, Large scale Bayesian inference and experimental design for sparse linear models, SIAM J. Imaging Sci., 4(1), 2011, pp. 166–199.

[23] M. Seeger and P. Wipf, Variational Bayesian inference techniques, IEEE Mag. Signal Proc., 27(6), 2010, pp. 81-91.

[24] C. Bekas, E. Kokiopoulou, and Y. Saad, An estimator for the diagonal of a matrix, Applied Numerical Mathematics, 57(11), 2007, pp. 1214–1229.

[25] G. Papandreou and A. L. Yuille, Efficient variational inference in large-scale Bayesian compressed sensing, Proc. IEEE Workshop on Information Theory in Computer Vision and Pattern Recognition, 2011.

[26] R. M. Neal and G. E. Hinton, A view of the EM algorithm that justifies incremental, sparse, and other variants, Learning in Graphical Models, Springer, 1998, pp. 355–368.

[27] A. Gunawardana and W. Byrne, Convergence theorems for generalized alternating minimization procedures, J. Mach. Learn. Res., 6, 2005, pp. 2049–2073.

[28] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos, The variational approximation for Bayesian inference, IEEE Mag. Signal Proc., 25(6), 2008, pp. 131–146.

[29] J. Palmer, D. Wipf, K. Kreutz-Delgado, and B. Rao, Variational EM algorithms for non-Gaussian latent variable models, Proc. Advances in Neural Information Processing systems, 18, 2006, page 1059.

[30] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul, Introduction to variational methods for graphical models, Machine Learning, 37, 1999, pp.183-233.

[31] C. F. Jeff Wu, On the convergence properties of the EM algorithm, The Annals of Statistics, 11, 1983, pp. 95–103.

[32] S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.

[33] C. M. Bishop, Pattern Recognition and Machine Learning, 2006, Springer New York.

[34] W. I. Zhangwill, Nonlinear Programming: A Unified Approach, Prentice-Hall, 1969.

[35] R. Chartrand and W. Yin, Iteratively reweighted algorithms for compressive sensing, Proc. IEEE international conference on Acoustics, speech and signal processing, 2008, pp. 3869–3872.

[36] D. Wipf and S. Nagarajan, Iterative reweighted ℓ1 and ℓ2 methods for finding sparse solutions, IEEE
Journal of Selected Topics in Signal Proc., 4(2), 2010, pp. 317–329.

[38] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2012, John Wiley & Sons.

[39] E. Challis and D. Barber, *Concave Gaussian variational approximations for inference in large-scale Bayesian linear models*, Proc. International Conference on Artificial Intelligence and Statistics, 6, 2011, page 7.

[40] M. E. Khan, A. Aravkin, M. Friedlander, and M. Seeger, *Fast dual variational inference for non-conjugate latent Gaussian models*, Proc. International Conference on Machine Learning, 2013, pp. 951–959.

[41] D. C. Sorensen, “Newton’s method with a model trust region modification,” SIAM J. Numerical Analysis, 19 (2), 1982, pp. 409–426.

[42] D. J. Crotty, R. L. McKinley, and M. P. Tornai, *Experimental spectral measurements of heavy k-edge filtered beams for x-ray computed mammotomography*, Phys. Med. Biol. 52, 2007, pp. 603.

[43] C. W. Dodge, *A rapid method for the simulation of filtered X-ray spectra in diagnostic imaging systems* (ProQuest, 2008).