Generalized Approximate Message Passing

Oxvig, Christian Schou; Arildsen, Thomas; Larsen, Torben

DOI (link to publication from Publisher):
10.5278/VBN.GAMPTechReport

Publication date:
2017

Document Version
Publisher's PDF, also known as Version of record

Link to publication from Aalborg University

Citation for published version (APA):
Oxvig, C. S., Arildsen, T., & Larsen, T. (2017). Generalized Approximate Message Passing: Relations and Derivations. DOI: 10.5278/VBN.GAMPTechReport

Generalized Approximate Message Passing
Relations and Derivations

Tech Report

Christian Schou Oxvig, Thomas Arildsen, Torben Larsen

Department of Electronic Systems
Aalborg University
Denmark

April 6, 2017

Generalized Approximate Message Passing: Relations and Derivations
Tech Report
Version: 1.00, April 6, 2017
doi:10.5278/VBN.GAMPTechReport

Christian Schou Oxvig, Thomas Arildsen, Torben Larsen
Department of Electronic Systems, Aalborg University, Denmark

Author Contributions

Christian Schou Oxvig defined the research which led to the novel contributions presented in this tech report and was the main contributor to the proposed solutions. He defined and detailed the outline of the GAMP elements to include in the review parts of the tech report. Finally, he wrote the entire draft of the tech report with the exception of Section 3.3 "MMSE Channel Functions in General" and revised the draft according to comments from co-authors.

Thomas Arildsen and **Torben Larsen** supervised the work on the novel contributions presented in this tech report. They contributed several ideas that were used in the proposed solutions. They took part in several discussions of the outline of the review parts of the tech report. Finally, these authors performed several reviews of drafts of the tech report and proposed several changes that were used in the final version of the tech report. Torben Larsen wrote Section 3.3 "MMSE Channel Functions in General".

Contents

1 Introduction

This tech report details a collection of results related to the Generalised Approximate Message Passing (GAMP) [1] algorithm. It is a summary of the results that the authors have found critical in understanding the GAMP algorithm. In particular, emphasis is on the details that are crucial in implementing the GAMP algorithm on a computer but which are oftentimes left out from the literature focusing on the more theoretical aspects of GAMP. Thus, this tech report is not meant to comprehensively cover all the published works on GAMP and related algorithms.

The Generalised Approximate Message Passing (GAMP) [1] algorithm by Rangan is a generalisation of Approximate Message Passing (AMP) algorithm, independently described by Donoho et al. [2], [3], [4] and Krzakala et al. [5], [6]. The generalisation allows for arbitrary output channels to be used with AMP.

1.1 Contributions Overview

The primary focus of this tech report is on giving more elaborate derivations and discussions of GAMP key relations found in the existing literature - with an emphasis on implementation details. However, a few new results are presented. Specifically, our new contributions are:

1. The proposed General Weighted Sparse (GWS) GAMP input channel described in Section 3.5 and all of its related derivations including the EM updates described in Sections 6.2.2.2 and 6.2.2.3 and the relations described in Sections 3.7.1.2 and 3.7.2.1.

2. The Sparse Bernoulli-Laplace input channel derivations given in Section 3.7.1 and the related EM update derivations given in Sections 6.2.2.4 and 6.3.4.

3. The methods for efficiently computing the Frobenius norm of various system matrices described in Section 4.2.2.

1.2 Background

We consider the undersampling reconstruction problem of estimating $\boldsymbol{\alpha} \in \mathbb{C}^{n\times 1}$ from the measurements $\mathbf{y} \in \mathbb{C}^{m\times 1}$ when $m \leq n$ (typically $m \ll n$) and $\mathbf{y} = \mathbf{A}\boldsymbol{\alpha} + \mathbf{e}$ with:

- $\mathbf{A} \in \mathbb{C}^{m\times n}$ - the system matrix

- $\mathbf{e} \in \mathbb{R}^{m\times 1}$ - an additive measurement noise

The system matrix $\mathbf{A}$ may be further decomposed into the matrix product of the sampling operator matrix $\boldsymbol{\Phi} \in \mathbb{R}^{m\times p}$ and the dictionary matrix $\boldsymbol{\Psi} \in \mathbb{C}^{p\times n}$, i.e. $\mathbf{A} = \boldsymbol{\Phi}\boldsymbol{\Psi}$. The dictionary matrix is chosen such that the coefficient vector $\boldsymbol{\alpha}$ has some sort of *structure*, e.g. $\boldsymbol{\alpha}$ is sparse. Introducing the *signal of interest*, $\mathbf{x} \in \mathbb{C}^{p\times 1}$, as well as the noiseless measurements, $\mathbf{z} \in \mathbb{C}^{m\times 1}$, the setup is described by the following set of equations:

$$\begin{aligned}
\mathbf{y} &= \mathbf{A}\boldsymbol{\alpha} + \mathbf{e} && (1.1)\\
&= \mathbf{z} + \mathbf{e} && (1.2)\\
&= \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{\alpha} + \mathbf{e} && (1.3)\\
&= \boldsymbol{\Phi}\mathbf{x} + \mathbf{e} && (1.4)\\
\mathbf{x} &= \boldsymbol{\Psi}\boldsymbol{\alpha} && (1.5)\\
\mathbf{z} &= \mathbf{A}\boldsymbol{\alpha} && (1.6)
\end{aligned}$$

1.3 Motivation

For probabilistic inference/recovery/reconstruction problems as those described in Section 1.2, one can generally observe a *phase transition* [7] (in the large system limit $n \to \infty$ for fixed $\delta = m/n$) separating the problems for which successful inference can be done from those for which it is impossible [5], [6]. This separation is determined by the available information about the inference problem. If too little information is available, it is generally not possible to solve the inference problem. The overall goal is then to find algorithms that are able to reach the phase transition boundary, i.e., they must be able to solve the inference problems when provided with just enough information to be able to succeed. The AMP algorithm is of interest in this regard primarily due to [5], [8]:

- It finds Bayes optimal (maximum a posterior (MAP) or minimum mean squared error (MMSE)) estimates in probabilistic inference/recovery/reconstruction problems and is (under certain conditions) able to reach the phase transition boundary.

- The number of messages in the underlying message passing problem that is being approximated scales linearly with the problem size. Thus, it is a computationally feasible algorithm.

See e.g. [5] or [8] for a more detailed motivation.

1.4 Derivation

Here we give a brief overview of a more heuristic derivation of the AMP following [5] and [9]. Rigorous proofs are given in [10], [11].

1.4.1 MMSE and MAP GAMP

We consider a Bayesian probabilistic approach to reconstructing a vector $\boldsymbol{\alpha}$ from measurements $\mathbf{y}$ in a setup described by the set of Equations (1.1)-(1.6). In particular, we consider the posterior distribution:

$$p(\boldsymbol{\alpha}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})}{p(\mathbf{y})} \tag{1.7}$$

$$= \frac{1}{\mathcal{Z}} p(\mathbf{y}|\boldsymbol{\alpha})p(\boldsymbol{\alpha}) \tag{1.8}$$

for a normalisation constant $\mathcal{Z}$, the likelihood $p(\mathbf{y}|\boldsymbol{\alpha})$, and $p(\boldsymbol{\alpha})$ - a prior on $\boldsymbol{\alpha}$. As is usually the case in Bayesian methods, it all boils down to finding ways to handle the otherwise intractable integretions/summations needed to determine the value of such normalisation constants as well as marginals or expectations.

Even though the posterior marginals of $\boldsymbol{\alpha}$ may be of interest as such, one usually finds point estimates from these marginals. In particular, the minimum mean squared error (MMSE) or the maximum a posteriori (MAP) estimates are often used:

$$\alpha_j^{\text{MMSE}} = \int_{\alpha_j} \alpha_j p(\alpha_j|\mathbf{y}) d\alpha_j \tag{1.9}$$

$$\alpha_j^{\text{MAP}} = \arg\max_{\alpha_j} p(\alpha_j|\mathbf{y}) \tag{1.10}$$

where the (conditional) marginals are obtained as

$$p(\alpha_j|\mathbf{y}) = \int_{\{\alpha_{j'}\}:j' \neq j} p(\boldsymbol{\alpha}|\mathbf{y}) d\boldsymbol{\alpha} \tag{1.11}$$

Thus, we are interested in finding MMSE (or MAP) estimates of $\boldsymbol{\alpha}$ which entails the need for (at least indirectly) finding the marginals. In rest of this tech report, we generally focus on GAMP for finding MMSE estimates.

1.4.2 The Message Passing Interpretation

In general, the system matrix $\mathbf{A}$ defines a dense bipartite factor graph with variables $\alpha_1, \dots, \alpha_n$ and factors $y_1, \dots, y_m$ [9]. Had this factor graph contained no loops, it would have been possible to obtain exact inference of the marginals in Equation (1.11) (and MMSE estimates) in a single pass over the graph by use of the sum-product message passing algorithm (sometimes also known as belief propagation) [12], [13]. However, due to the loops in the factor graph, iterative message passing over the graph only results in approximate inference. In the loopy sum-product message passing, one iteratively passes the following messages (full probability distributions) along the edges of the factor graph:

$$m_{i\to j}(\alpha_j) = \frac{1}{\mathcal{Z}^{i\to j}} \int_{\{\alpha_k\}:k\neq j} p(y_i|z_i) \prod_{k\neq j} m_{k\to i}(\alpha_k) \qquad \text{factor to variable message} \tag{1.12}$$

$$m_{j\to i}(\alpha_j) = \frac{1}{\mathcal{Z}^{j\to i}} p(\alpha_j|[\boldsymbol{\theta}_I]_j) \prod_{l\neq i} m_{l\to j}(\alpha_j) \qquad \text{variable to factor message} \tag{1.13}$$

where the $z_i = [\mathbf{A}\alpha]_i$'s are the noiseless measurements, $[\boldsymbol{\theta}_I]_j$ are some parameters of the prior distribution on α_j and $\mathcal{Z}^{i\to j}$, $\mathcal{Z}^{j\to i}$ are normalisation factors. The use of this iterative message passing scheme poses two problems:

1. One has to track full probability distributions (the messages $m_{i\to j}(\alpha_j)$, $m_{j\to i}(\alpha_j)$ are full probability distributions on α_j, i.e. functions on the real axis).

2. There is a total of $2mn$ messages that must be passed in each iteration (all m factors sends a message to all n variables and vice versa).

Thus, it is intractable to use the above message passing scheme as is. However, under certain assumptions, it is possible to use a different message passing scheme involving only $m+n$ messages. Furthermore, these messages are means and variances, i.e, they are scalars which provides for a much more tractable algorithm.

The derivation of such a message passing scheme relies on the assumption that the individual element of $\mathbf{A}$ becomes insignificant for $n \to \infty$. That is, it is assumed that all elements of $\mathbf{A}$ scale like $\frac{1}{\sqrt{n}}$ such that each element seen in isolation becomes insignificant in the large system limit $n \to \infty$. In other words, the information is spread equally throughout the graph. The use of various Taylor approximations and applications of the central limit theorem then gives the resulting new message passing scheme. All the details are given in [2], [3], [4], [5], [6], [9], [14], [15].

Most importantly from an implementation perspective, the workload reduces to the iteration of the mean and variance updates $\bar{\alpha}_j$ and $\tilde{\alpha}_j$ (along with a few other states as detailed below) for finding the MMSE estimate[1] of α_j, $j = 1, \dots, n$

$$\bar{\alpha}_j = f_{\bar{\alpha}_j}(s_j, r_j; [\boldsymbol{\theta}_I]_j) \tag{1.14}$$

$$\tilde{\alpha}_j = f_{\tilde{\alpha}_j}(s_j, r_j; [\boldsymbol{\theta}_I]_j) \tag{1.15}$$

for scalar non-linear functions $f_{\bar{\alpha}_j}$, $f_{\tilde{\alpha}_j}$. These updates may be iterated to a fixed point resulting in the reconstructed signal $\bar{\boldsymbol{\alpha}} = [\bar{\alpha}_1, \dots, \bar{\alpha}_n]^T$. The estimated variance of the elements of $\bar{\boldsymbol{\alpha}}$ is then given by $\tilde{\boldsymbol{\alpha}}$ [16]. This variance may be used to quantify the accuracy of the reconstructed signal (it should be small). In summary, $\bar{\alpha}_j$ ends up being an approximation of α_j^{MMSE} in Equation (1.9) with $\tilde{\alpha}_j$ expressing something about the quality of this approximation.

As hinted by the scalar functions in Equations (1.14) and (1.15), one introduces additional states that represent local beliefs about means and variances at the variables and factors:

s_j: Prior side (/ AMP field / variable field) variance

r_j: Prior side mean

v_i: Factor side (/ Channel side / factor field) variance

[1] The GAMP framework of Rangan [1], [16] allows for both MMSE and MAP estimates by choosing different channel functions.

o_i: Factor side mean

Note that there is one such state for each $j = 1, \dots, n$, $i = 1, \dots, m$. Thus, the number of states in the AMP algorithm that must be tracked and updated in each iteration is $\mathcal{O}(m+n)$. Only the compact (and simplified) AMP iteration by Donoho/Maleki/Montanari has exactly $m+n$ messages that must be tracked in each iteration [2] - see also Section 2.1. The full GAMP iteration is given in Section 2 (Equations (2.1) - (2.12)). The generalisation in GAMP introduces further states as well as a pair of scalar output side channel functions $f_{\bar{z}_i}$, $f_{\tilde{z}_i}$ in addition to the input side channel functions $f_{\bar{\alpha}_j}$, $f_{\bar{\alpha}_j}$. Details about these channel functions are given in Section 3.

1.4.3 State Evolution

From a theoretical perspective, the probably most appealing part of the AMP algorithm is its State Evolution (SE) formalism [2], [4], [9] which provides precise convergence guarantees for AMP. It turns out that in the large system limit $n \to \infty$ and under certain conditions, a certain state (the unthresholded $\hat{\boldsymbol{\alpha}}$) in the AMP algorithm may be interpreted as a AWGN corrupted version of the true $\boldsymbol{\alpha}$ for any iteration. Thus, the reconstruction error may be tracked by the mean squared error (MSE) of the estimate. This is, however, a purely analytical construction that may be used to theoretically characterise the AMP algorithm in the large system limit for a given problem. A rigorous proof of the SE is given in [10].

1.4.4 Theoretical Guarantees for Arbitrary System Matrices

The AMP algorithm has been rigorously proved to converge for i.i.d. Gaussian system matrices (entries are drawn i.i.d. zero-mean Gaussian) in the large system limit $n \to \infty$ [17], [18] as well as for i.i.d. sub-Gaussian system matrices [10], [11]. For i.i.d. Gaussian system matrices of finite size, it can be shown that the probability of deviation from the SE described in Section 1.4.3 decreases exponentially in n [19].

For various other types of system matrices, a damping strategy may be used to guarantee convergence of the GAMP algorithms [20] (more details are given in Section 5.2). The S-AMP algorithm described in [21] and [22] is an attempt at generalising GAMP to more general system matrix ensembles. A similar attempt at an AMP algorithm for more general system matrix ensembles is the ADMM-GAMP algorithm [23]. Yet another attempt is the Orthogonal AMP [24] algorithm which is somewhat similar to the Vector AMP algorithm[2]. Generally, these alternatives have significantly higher computationally complexity than GAMP. Thus, they all present a trade-off between convergence guarantee and computational complexity.

The fixed points of GAMP with arbitrary matrices is discussed in [25] whereas general compressed sensing phase transitions for deterministic matrices are discussed in [26]. Details about when AMP algorithms provide theoretically optimal recovery guarantees are given in [6], [27]. Finally, empirical results suggest that GAMP also converges for various other system matrices [10], [28] including matrices related to structured random matrices [29] and structurally random matrices [30].

1.4.5 Additional Practicality Notes on GAMP

The AMP algorithm is based on a zero mean assumption on $\mathbf{A}$, i.e., the entries of $\mathbf{A}$ are assumed to be zero mean. For non-zero mean $\mathbf{A}$, one may use a transformation to a new problem with a zero mean system matrix, see e.g. [5] or [31].

In the derivation of AMP, Krzakala et al. assume a $\frac{1}{\sqrt{n}}$ scaling of the entries of $\mathbf{A}$ [6] whereas Montanari et. al assume a $\frac{1}{\sqrt{m}}$ with a fixed $\delta = \frac{m}{n}$ [9]. Theoretically, the two assumptions are equivalent (in the sense of an element becoming insignificant) for $n \to \infty$ as long as m scales linearly with n, i.e., δ is fixed. However, in practice (with finite n) for low δ, the difference between m and n is large enough to impact the convergence of the AMP algorithm[3].

[2] See the pre-print available at `https://arxiv.org/abs/1610.03082`

[3] Empirical phase transition simulations reported on in a submitted but yet to be published manuscript by the authors of this tech report confirms this observation. See also Chapter 4 for a more elaborate discussion of the impact of these assumptions.

The in- and output channels are required to be separable. That is, the channel defining probabilities must be conditionally independent [1]:

$$p(\boldsymbol{\alpha}|\boldsymbol{\theta}_I) = \prod_j p(\alpha_j|[\boldsymbol{\theta}_I]_j) \tag{1.16}$$

$$p(\mathbf{y}|\mathbf{z};\boldsymbol{\theta}_o) = \prod_i p(y_i|z_i;[\boldsymbol{\theta}_o]_i) \tag{1.17}$$

Thus, we consider a setup in which we imagine that a random $\boldsymbol{\alpha}$ is generated according to the input channel specification in Equation (1.16). This $\boldsymbol{\alpha}$ is measured through the linear transform in Equation (1.6). The observation $\mathbf{y}$ is then generated from $\mathbf{z}$ according to the output channel specification in Equation (1.17) which is a generalisation of the additive noise used in Equation (1.1). That being said, one may consider the $\boldsymbol{\theta}_I$'s and $\boldsymbol{\theta}_o$'s to be random variables themselves and define hyperpriors on them. This allows for arbitrary structures (subject to the above separability constraint) on the prior of $\boldsymbol{\alpha}$, e.g. a Markov chain prior [32]. For updating the beliefs across such priors, one may use the Turbo GAMP framework [33]. See also Chapter 8 for more references to works on structured priors. Independently of the choice of prior, it is important to keep in mind that the GAMP estimates, when MMSE- or MAP-optimal, are optimal under the model assumption which may only approximate the problem attempted solved.

1.5 Notation

The notation used across publications by a given author is typically consistent. Unfortunately, notation varies between different authors. Table 1.1 gives a comparison of the different notations used by some of the authors responsible for a significant part of the published works on GAMP. Also included in the table is our "unified" notation used in this note. This notation is a combination of the other notations as well as elements from the notation that is customary in the compressed sensing literature with a focus on imaging applications.

Quantity	Our notation	Krzakala	Schniter	Rangan	Donoho
System matrix	$\mathbf{A}$	$\mathbf{F}$	$\mathbf{A}$	$\mathbf{A}$	$\mathbf{A}$
Abs. entrywise squared $\mathbf{A}$	$\lvert\mathbf{A}\rvert^{\circ 2}$ *or* $\mathbf{A}_{\text{asq}}$			$\lvert\mathbf{A}\rvert^{2}$	
Dictionary coefficients	$\boldsymbol{\alpha}$	$\mathbf{x}$ *or* $\mathbf{s}$	$\mathbf{x}$	$\mathbf{x}$ *or* $\mathbf{s}$	$\mathbf{x}$ *or* $\mathbf{s}$
Noiseless measurements	$\mathbf{z}$		$\mathbf{z}$	$\mathbf{z}$	
Noisy measurements	$\mathbf{y}$	$\mathbf{y}$	$\mathbf{y}$	$\mathbf{y}$	$\mathbf{y}$
Additive noise	$\mathbf{e}$	$\boldsymbol{\xi}$	$\mathbf{w}$	$\mathbf{w}$	$\mathbf{w}$
Image as a vector	$\mathbf{x}$				
Image width	w				
Image height	h				
Image as a matrix	$\mathbf{M}$				
Number of measurements	m	M	m *or* M	m	n
Number of coefficients	n	N	n *or* N	n	N
Number of non-zeros	k	K	k *or* K		k
Undersampling ratio	$\delta = \frac{m}{n}$	α			δ
Sparsity level	$\rho = \frac{k}{m}$				ρ
Signal density	$\tau = \frac{k}{n}$	ρ	λ	ρ	
Dictionary	$\boldsymbol{\Phi}$				
Sampling matrix	$\boldsymbol{\Psi}$				
Input (/prior) parameters	$\boldsymbol{\theta}_I$		$\mathbf{q}$	$\mathbf{q}$	
Output parameters	$\boldsymbol{\theta}_o$				
Factor-side state #1	$\mathbf{v}$ / v	$\mathbf{V}$ / $\bar{V}$	$\boldsymbol{\mu}^p$ / μ^p	$\boldsymbol{\tau}^p$ / τ^p	γ
Factor-side state #2	$\mathbf{o}$	$\boldsymbol{\omega}$	$\hat{\mathbf{p}}$	$\hat{\mathbf{p}}$	
Channel function	f	f		g	F/G *or* (η)[‡]
Factor-side mean	$\bar{\mathbf{z}}$		$\hat{\mathbf{z}}$		
Factor-side variance	$\tilde{\mathbf{z}}$		$\boldsymbol{\mu}^z$		
Output channel state #1	$\mathbf{q}$		$\hat{\mathbf{s}}$	$\hat{\mathbf{s}}$	
Output channel state #2	$\mathbf{u}$ / u		$\boldsymbol{\mu}^s$ / μ^s	$\boldsymbol{\tau}^s$ / τ^s	
Variable-side state #1	$\mathbf{s}$ / s	$\boldsymbol{\Sigma}^2$ / Σ^2	$\boldsymbol{\mu}^r$ / μ^r	$\boldsymbol{\tau}^r$ / τ^r	
Variable-side state #2	$\mathbf{r}$	$\mathbf{R}$	$\hat{\mathbf{r}}$	$\hat{\mathbf{r}}$	
Variable-side mean	$\bar{\boldsymbol{\alpha}}$	$\mathbf{a}$	$\hat{\mathbf{x}}$	$\hat{\mathbf{x}}$	$\mathbf{x}$
Variable-side variance	$\tilde{\boldsymbol{\alpha}}$	$\mathbf{v}$	$\boldsymbol{\mu}^x$	$(\boldsymbol{\tau}^x)$[†]	
Onsager-corrected residual	$\boldsymbol{\chi}$		$\hat{\mathbf{v}}$		$\mathbf{z}$
(Marginal) prob. density	$p(x)$	$\phi(x)$	$p_X(x)$	$p_X(x)$	
Joint prob. density	$p(x, y)$	$P(x, y)$	$p_{X,Y}(x, y)$	$p_{X,Y}(x, y)$	
Conditional prob. density	$p(x\|y;\theta)$	$P(x\|y,\theta)$	$p_{X\|Y;\theta}(x\|y;\theta)$		
Normalisation factor	$\mathcal{Z}$	Z	Z	Z	Z
AWGN noise variance	σ^2	Δ	μ^w *or* ψ	τ^w	σ^2
Gaussian mean	$\bar{\theta}$	$\bar{x}$	$\hat{\theta}$ *or* θ	q	
Gaussian variance	$\tilde{\theta}$	σ^2	μ^θ *or* ϕ	τ^{x0}	
Laplacian rate parameter	λ		λ		β
Laplacian mean parameter	μ				
Convergence tolerance	ϵ		τ_{gamp}		
Iteration index	t		t	t	t
Maximum iterations	$T_{\max}$		$T_{\max}$		
Step-size parameter	κ		β		
Dirac delta	δ_{Dirac}	δ	δ		δ

[‡] Only under certain conditions is the η threshold functions equivalent to the GAMP channel functions - see Section 2.1.

[†] Rangan uses slightly different states to allow for both MMSE and MAP estimates. See [1], [16], [20].

Table 1.1: Comparison of notations typically used by the different research groups working on (G)AMP.

2 The GAMP Iteration

The AMP iteration is given in [5]. Here we state the MMSE Generalised AMP (GAMP) iteration following Parker's presentation [15]. Rangan was the first to state the GAMP iteration [1]. However, Rangan uses special output functions which allow for both MMSE and MAP estimates. The relation between the below MMSE GAMP iteration and Rangan's more general GAMP iteration is elaborated on in Section 3. The optional use of parameter value updates is described in [5] and [28]. The MMSE GAMP iteration consists of the state updates in Equations (2.1)-(2.12).

Output (factor) side updates:

$$v_i^{t+1} = \sum_j |a_{ij}|^2 \tilde{\alpha}_j^t \tag{2.1}$$

$$o_i^{t+1} = \sum_j a_{ij} \bar{\alpha}_j^t - v_i^{t+1} q_i^t \tag{2.2}$$

$$\bar{z}_i^{t+1} = f_{\bar{z}_i}(v_i^{t+1}, o_i^{t+1}; y_i, [\boldsymbol{\theta}_o]_i^t) \tag{2.3}$$

$$\tilde{z}_i^{t+1} = f_{\tilde{z}_i}(v_i^{t+1}, o_i^{t+1}; y_i, [\boldsymbol{\theta}_o]_i^t) \tag{2.4}$$

$$q_i^{t+1} = \frac{\bar{z}_i^{t+1} - o_i^{t+1}}{v_i^{t+1}} \tag{2.5}$$

$$u_i^{t+1} = \frac{v_i^{t+1} - \tilde{z}_i^{t+1}}{(v_i^{t+1})^2} \tag{2.6}$$

Input (variable) side updates:

$$s_j^{t+1} = \left[\sum_i |a_{ij}|^2 u_i^{t+1} \right]^{-1} \tag{2.7}$$

$$r_j^{t+1} = \bar{\alpha}_j^t + s_j^{t+1} \sum_i a_{ij}^* q_i^{t+1} \tag{2.8}$$

$$\bar{\alpha}_j^{t+1} = f_{\bar{\alpha}_j}(s_j^{t+1}, r_j^{t+1}; [\boldsymbol{\theta}_I]_j^t) \tag{2.9}$$

$$\tilde{\alpha}_j^{t+1} = f_{\tilde{\alpha}_j}(s_j^{t+1}, r_j^{t+1}; [\boldsymbol{\theta}_I]_j^t) \tag{2.10}$$

Optional parameter value updates (using e.g. EM - see also Section 6):

$$[\boldsymbol{\theta}_o]_i^{t+1} = \dots \tag{2.11}$$

$$[\boldsymbol{\theta}_I]_j^{t+1} = \dots \tag{2.12}$$

where a_{ij}^* is the complex conjugate of a_{ij}.

2.1 Relation to Donoho/Maleki/Montanari AMP

Rangan states that GAMP [16] is closely related to the AMP algorithm by Donoho/Maleki/Montanari [2], [3], [4], [9], [34]. Parker [15] gives the below elaboration on this claim in relation to the MMSE GAMP (see also [35]).

We consider the MMSE GAMP with the AWGN output channel given in Equations (3.57) and (3.60) (used in computing $\bar{z}$ and $\tilde{z}$). The equivalence to the DMM AMP is based on a few simplifications of some of the GAMP states. In particular, v, u, and s become scalars (which do not depend on the index j)

$$v^{t+1} := \frac{1}{m}\sum_j \tilde{\alpha}_j^t \approx \sum_j |a_{ij}|^2 \tilde{\alpha}_j^t \tag{2.13}$$

$$u^{t+1} := \frac{v^{t+1} - \tilde{z}^{t+1}}{(v^{t+1})^2} = \frac{v^{t+1} - \frac{\sigma^2 v^{t+1}}{\sigma^2 + v^{t+1}}}{(v^{t+1})^2} = \frac{\frac{v^{t+1}\sigma^2 + (v^{t+1})^2 - v^{t+1}\sigma^2}{\sigma^2 + v^{t+1}}}{(v^{t+1})^2} = \frac{1}{\sigma^2 + v^{t+1}} \tag{2.14}$$

$$s^{t+1} := \frac{1}{u^{t+1}} = \sigma^2 + v^{t+1} \approx \left[\frac{1}{m}\sum_i u_i^{t+1}\right]^{-1} \approx \left[\sum_i |a_{ij}|^2 u_i^{t+1}\right]^{-1} \tag{2.15}$$

These simplifications are closely related to the sum approximations by Krzakala et al. in Equations (4.7) and (4.8), though the scaling (i.e. the assumed variance of the the entries in $\mathbf{A}$) is slightly different: $\frac{1}{m}$ vs $\frac{1}{n}$. Based on the above simplifications, the update of the r_j state becomes

$$r_j^{t+1} = \bar{\alpha}_j^t + s^{t+1}\sum_i a_{ij}^* q_i^{t+1} \tag{2.16}$$

$$= \bar{\alpha}_j^t + s^{t+1}\sum_i a_{ij}^* \frac{\bar{z}_i^{t+1} - o_i^{t+1}}{v^{t+1}} \tag{2.17}$$

$$= \bar{\alpha}_j^t + s^{t+1}\sum_i a_{ij}^* \frac{o_i^{t+1} + \frac{v^{t+1}}{\sigma^2 + v^{t+1}}(y_i - o_i^{t+1}) - o_i^{t+1}}{v^{t+1}} \tag{2.18}$$

$$= \bar{\alpha}_j^t + s^{t+1}\sum_i a_{ij}^* \frac{y_i - o_i^{t+1}}{\sigma^2 + v^{t+1}} \tag{2.19}$$

$$= \bar{\alpha}_j^t + \sum_i a_{ij}^* (y_i - o_i^{t+1}) \tag{2.20}$$

$$= \bar{\alpha}_j^t + \sum_i a_{ij}^* \chi_i^{t+1} \tag{2.21}$$

for

$$\chi_i^{t+1} := y_i - o_i^{t+1} \tag{2.22}$$

Now, for χ_i^{t+1}, we have

$$\chi_i^{t+1} = y_i - o_i^{t+1} \tag{2.23}$$

$$= y_i - \left(\sum_j a_{ij}\bar{\alpha}_j^t - v^{t+1} q_i^t \right) \tag{2.24}$$

$$= y_i - \sum_j a_{ij}\bar{\alpha}_j^t + v^{t+1}\frac{\bar{z}_i^t - o_i^t}{v^t} \tag{2.25}$$

$$= y_i - \sum_j a_{ij}\bar{\alpha}_j^t + v^{t+1}\frac{y_i - o_i^t}{\sigma^2 + v^t} \tag{2.26}$$

$$= y_i - \sum_j a_{ij}\bar{\alpha}_j^t + \frac{v^{t+1}}{s^t}\chi_i^t \tag{2.27}$$

$$= y_i - \sum_j a_{ij}\bar{\alpha}_j^t + \frac{\frac{1}{m}\sum_j \tilde{\alpha}_j^t}{s^t}\chi_i^t \tag{2.28}$$

$$= y_i - \sum_j a_{ij}\bar{\alpha}_j^t + \frac{\frac{1}{m}\sum_j s^t g'_{in}(r_j^t, \boldsymbol{\theta}_I, s^t)}{s^t}\chi_i^t \tag{2.29}$$

$$= y_i - \sum_j a_{ij}\bar{\alpha}_j^t + \frac{1}{m}\sum_j g'_{in}(r_j^t, \boldsymbol{\theta}_I, s^t)\chi_i^t \tag{2.30}$$

$$= y_i - \sum_j a_{ij}\bar{\alpha}_j^t + \frac{n}{m}\langle g'_{in}(r_j^t, \boldsymbol{\theta}_I, s^t)\rangle\chi_i^t \tag{2.31}$$

$$= y_i - \sum_j a_{ij}\bar{\alpha}_j^t + \frac{1}{\delta}\langle g'_{in}(\bar{\alpha}_j^{t-1} + \sum_i a_{ij}^*\chi_i^t, \boldsymbol{\theta}_I, s^t)\rangle\chi_i^t \tag{2.32}$$

where we have used Rangan's GAMP g'_{in} channel in Equation (3.26) and $\langle\cdot\rangle$ denotes the average. Similarly for $\bar{\alpha}^{t+1}$, we have

$$\bar{\alpha}_j^{t+1} = f_{\bar{\alpha}_j}(s_j^{t+1}, r_j^{t+1}; \boldsymbol{\theta}_I) \tag{2.33}$$

$$= g_{in}(r_j^{t+1}, \boldsymbol{\theta}_I, s_j^{t+1},) \tag{2.34}$$

$$= g_{in}(\bar{\alpha}_j^t + \sum_i a_{ij}^*\chi_i^{t+1}, \boldsymbol{\theta}_I, s_j^{t+1}) \tag{2.35}$$

where we have used Rangan's GAMP g_{in} channel in Equation (3.25). Now in matrix-vector notation, Equations (2.32) and (2.35) read

$$\boldsymbol{\chi}_t = \mathbf{y} - \mathbf{A}\bar{\boldsymbol{\alpha}}_{t\text{-}1} + \frac{1}{\delta}\langle g'_{in}(\bar{\boldsymbol{\alpha}}_{t\text{-}2} + \mathbf{A}^H\boldsymbol{\chi}_{t\text{-}1}, \boldsymbol{\theta}_I, \mathbf{s}_{t\text{-}1})\rangle\boldsymbol{\chi}_{t\text{-}1} \tag{2.36}$$

$$\bar{\boldsymbol{\alpha}}_t = g_{in}(\bar{\boldsymbol{\alpha}}_{t\text{-}1} + \mathbf{A}^H\boldsymbol{\chi}_t, \boldsymbol{\theta}_I, \mathbf{s}_t) \tag{2.37}$$

where ^{H} denotes the Hermitian transpose (complex conjugated transpose), $\boldsymbol{\chi}$ is the so-called Onsager-corrected residual and $\mathbf{s}_t$ is a length m vector with all entries equal to the scalar in Equation (2.15). Here we note that from Equations (2.13) and(2.15), using similar steps as in deriving Equation (2.32), we have

$$v^{t+1} = \frac{1}{\delta}s^t\langle g'_{in}(\bar{\alpha}_j^{t-1} + \sum_i a_{ij}^*\chi_i^t, \boldsymbol{\theta}_I, s^t)\rangle \tag{2.38}$$

$$= \frac{1}{\delta}(\sigma^2 + v^t)\langle g'_{in}(\bar{\alpha}_j^{t-1} + \sum_i a_{ij}^*\chi_i^t, \boldsymbol{\theta}_I, \sigma^2 + v^t)\rangle \tag{2.39}$$

Thus, in order to compute the channel value based on s, one may introduce the additional recursion on the state v. If we take $\eta_t(\cdot) = g_{in}(\cdot, \boldsymbol{\theta}_I, \mathbf{s}_t)$ for the threshold function η_t in [2], then Equations

(2.36) and (2.37) constitute the Donoho/Maleki/Montanari AMP update in Equations [**2**] and [**1**], respectively, in [2][1].

The threshold function, η_t is in general a conditional expectation [3] (as is $g_{in}(\cdot, \boldsymbol{\theta}_I, \mathbf{s}_t)$) in MMSE GAMP. However, in AMP for the Basis Pursuit and LASSO problems [2], [3] (closely linked to instances of the MAP GAMP [1], [16], the threshold function becomes the soft threshold operator for which the threshold level is chosen slightly differently - see [9] for more details. See also [15] for a further discussion of some of these subtle details in the choice of threshold function.

2.2 Relation to IST, ISTA, and ADMM

The iterative soft thresholding (IST) algorithm [36] is similar in structure to the DMM AMP updates in Equations (2.36) and (2.37). Specifically, the corresponding IST updates read

$$\boldsymbol{\chi}_t = \mathbf{y} - \mathbf{A}\bar{\boldsymbol{\alpha}}_{t\text{-}1} \tag{2.40}$$

$$\bar{\boldsymbol{\alpha}}_t = \eta_t(\bar{\boldsymbol{\alpha}}_{t\text{-}1} + \mathbf{A}^H\boldsymbol{\chi}_t) \tag{2.41}$$

for η_t being the soft threshold operator. Thus, the difference to AMP is the lack of the Onsager correction $\frac{1}{\delta}\langle g'_{in}(\bar{\boldsymbol{\alpha}}_{t\text{-}2} + \mathbf{A}^H\boldsymbol{\chi}_{t\text{-}1}, \boldsymbol{\theta}_I, \mathbf{s}_{t\text{-}1})\rangle\boldsymbol{\chi}_{t\text{-}1}$. It is this correction that gives rise to the interpretation of $\bar{\boldsymbol{\alpha}}_{t\text{-}1} + \mathbf{A}^H\boldsymbol{\chi}_t$ being a AWGN corrupted version of the true $\boldsymbol{\alpha}$ as discussed in Section 1.4.3 [2].

GAMP may also be interpreted as certain variants of the iterative shrinkage and thresholding algorithm (ISTA) [37] and the alternating direction method of multipliers (ADMM) algorithm [38] as detailed in [25].

[1] In [2], $\bar{\boldsymbol{\alpha}}_t$ is computed prior to $\boldsymbol{\chi}_t$ which accounts for the iteration index shifts.

3 MMSE Channel Functions

The MMSE GAMP channel functions, $f_{\bar{\alpha}}$, $f_{\tilde{\alpha}}$, $f_{\bar{z}}$, $f_{\tilde{z}}$, used in Equations (2.1)-(2.12) are given in terms of special conditional expectations and variances since these are at the core of the MMSE GAMP as described in Section 1.4.1. The MMSE GAMP channel functions follow the AMP channel function definitions by Krzakala et. al in [5] but differs from GAMP the channel function definitions used by Rangan [1] as elaborated on in Section 3.3.1. All channel functions are scalar functions. Thus, in this chapter we drop the notational dependence on the index as well as the notational dependence on iteration. When the presented channel expressions are used with the GAMP iteration in Equations (2.1)-(2.12), the appropriate dependencies should be taken into account.

3.1 Input Side Channel Functions

The GAMP input side channel functions are:

$$f_{\bar{\alpha}}(s, r; \boldsymbol{\theta}_I) = \mathbb{E}_{\alpha|s,r,\boldsymbol{\theta}_I}[\alpha] \coloneqq \frac{1}{\mathcal{Z}_I} \int_\alpha \alpha p(\alpha; \boldsymbol{\theta}_I) \mathcal{N}(\alpha; r, s) d\alpha \tag{3.1}$$

$$f_{\tilde{\alpha}}(s, r; \boldsymbol{\theta}_I) = \mathrm{Var}_{\alpha|s,r,\boldsymbol{\theta}_I}(\alpha) \coloneqq \frac{1}{\mathcal{Z}_I} \int_\alpha |\alpha|^2 p(\alpha; \boldsymbol{\theta}_I) \mathcal{N}(\alpha; r, s) d\alpha - |f_{\bar{\alpha}}(s, r; \boldsymbol{\theta}_I)|^2 \tag{3.2}$$

where $\mathcal{Z}_I = \int_\alpha p(\alpha; \boldsymbol{\theta}_I) \mathcal{N}(\alpha; r, s) d\alpha$ is a normalisation constant that ensures that the product $p(\alpha; \boldsymbol{\theta}_I) \mathcal{N}(\alpha; r, s)$ is a proper probability measure and

$$\mathcal{N}(\alpha; r, s) = \frac{1}{\sqrt{2\pi s}} \exp\left(-\frac{1}{2} \frac{(\alpha - r)^2}{s}\right) \tag{3.3}$$

Thus, from Equation (3.1) we find that in GAMP, the true marginal posterior $p(\alpha|\mathbf{y}; \boldsymbol{\theta}_I)$ is approximated by:

$$p(\alpha|\mathbf{y}; s, r, \boldsymbol{\theta}_I) \coloneqq \frac{p(\alpha; \boldsymbol{\theta}_I) \mathcal{N}(\alpha; r, s)}{\int_\alpha p(\alpha; \boldsymbol{\theta}_I) \mathcal{N}(\alpha; r, s) d\alpha} \tag{3.4}$$

which has the interpretation that if $\tilde{\mathcal{A}}$ is a random variable distributed according to $p(\alpha; \boldsymbol{\theta}_I)$ and $\tilde{\mathcal{B}} = \tilde{\mathcal{A}} + \mathcal{W}$ with $\mathcal{W}$ a zero-mean Gaussian noise with variance s, then Equation (3.1) is the conditional mean of $\tilde{\mathcal{A}}$ given $\tilde{\mathcal{B}} = r$, i.e. $\mathbb{E}[\tilde{\mathcal{A}}|\tilde{\mathcal{B}} = r]$. Similarly Equation (3.2) is the conditional variance $\mathrm{Var}(\tilde{\mathcal{A}}|\tilde{\mathcal{B}} = r)$ [3], [16].

Note that the input channel parameters $\boldsymbol{\theta}_I$ may depend on the coefficient, i.e. the $[\boldsymbol{\theta}_I]_j$'s may be different for each $j = 1, \ldots, n$. In Equation (3.4) it is to be understood that $\boldsymbol{\theta}_I$ is the vector/matrix of *all* input channel parameters independently of whether or not they depend on the index j.

All the input channel functions are scalar functions. When used with vectors as arguments, it is to be understood that a channel function is used on each element of the vector.

3.2 Output Side Channel Functions:

The GAMP output side channel functions are:

$$f_{\bar{z}}(v, o; y, \boldsymbol{\theta}_o) = \mathbb{E}_{z|o,v,y,\boldsymbol{\theta}_o}[z] \coloneqq \frac{1}{\mathcal{Z}_o} \int_z z p(y|z; \boldsymbol{\theta}_o) \mathcal{N}(z; o, v) dz \tag{3.5}$$

$$f_{\tilde{z}}(v, o; y, \boldsymbol{\theta}_o) = \mathrm{Var}_{z|o,v,y,\boldsymbol{\theta}_o}(z) \coloneqq \frac{1}{\mathcal{Z}_o} \int_z |z|^2 p(y|z; \boldsymbol{\theta}_o) \mathcal{N}(z; o, v) dz - |f_{\bar{z}}(v, o; y, \boldsymbol{\theta}_o)|^2 \tag{3.6}$$

where $\mathcal{Z}_o = \int_z p(y|z;\boldsymbol{\theta}_o)\mathcal{N}(z;o,v)dz$ is a normalisation constant that ensures that the product $p(y|z;\boldsymbol{\theta}_o)\mathcal{N}(z;o,v)$ is a proper probability measure and

$$\mathcal{N}(z;o,v) = \frac{1}{\sqrt{2\pi v}}\exp\left(-\frac{1}{2}\frac{(z-o)^2}{v}\right) \tag{3.7}$$

Thus, from Equation (3.5) we find that in GAMP, the true marginal posterior $p(z|\mathbf{y};\boldsymbol{\theta}_o)$ is approximated by:

$$p(z|\mathbf{y};o,v,\boldsymbol{\theta}_o) \coloneqq \frac{p(y|z;\boldsymbol{\theta}_o)\mathcal{N}(z;o,v)}{\int_z p(y|z;\boldsymbol{\theta}_o)\mathcal{N}(z;o,v)dz} \tag{3.8}$$

which has the interpretation that if $\tilde{\mathcal{Y}}$ is a random variable distributed according to $p(y|z;\boldsymbol{\theta}_o)$ and $\tilde{\mathcal{Z}}$ is a Gaussian random variable with mean o and variance v, then Equation (3.5) is the conditional mean of $\tilde{\mathcal{Z}}$ given $\tilde{\mathcal{Y}} = y$, i.e. $\mathbb{E}[\tilde{\mathcal{Z}}|\tilde{\mathcal{Y}} = y]$. Similarly, Equation (3.6) is the conditional variance $\mathrm{Var}(\tilde{\mathcal{Z}}|\tilde{\mathcal{Y}} = y)$ [16].

Note that the output channel parameters $\boldsymbol{\theta}_o$ may depend on the coefficient, i.e. the $[\boldsymbol{\theta}_o]_i$'s may be different for each $i = 1,\dots,m$. In Equation (3.8), it is to be understood that $\boldsymbol{\theta}_o$ is the vector/matrix of *all* output channel parameters independently of whether or not they depend on the index i.

All the output channel functions are scalar functions. When used with vectors as arguments, it is to be understood that a channel function is used on each element of the vector.

3.3 MMSE Channel Functions in General

From Equations (3.1) and (3.2) as well as Equations (3.5) and (3.6), we find that in the MMSE case, evaluating the channels functions amounts to evaluating conditional means and variances. Thus, in relation to the in- and output channel functions, we may, for a given channel distribution $p(u;\boldsymbol{\theta})$, in general, define

$$\mathcal{Z}(v,w,\boldsymbol{\theta}) \coloneqq \int_u p(u;\boldsymbol{\theta})\mathcal{N}(u;v,w)du \in \mathbb{R} \tag{3.9}$$

$$N_1(v,w,\boldsymbol{\theta}) \coloneqq \int_u up(u;\boldsymbol{\theta})\mathcal{N}(u;v,w)du \in \mathbb{C} \tag{3.10}$$

$$N_{2a}(v,w,\boldsymbol{\theta}) \coloneqq \int_u |u|^2 p(u;\boldsymbol{\theta})\mathcal{N}(u;v,w)du \in \mathbb{C} \tag{3.11}$$

for which we may find mean and variance functions as

$$\mathbb{E}_{u|v,w,\boldsymbol{\theta}}[u] = \frac{N_1(v,w,\boldsymbol{\theta})}{\mathcal{Z}(v,w,\boldsymbol{\theta})} \in \mathbb{C} \tag{3.12}$$

$$\mathrm{Var}_{u|v,w,\boldsymbol{\theta}}(u) = \frac{N_{2a}(v,w,\boldsymbol{\theta})}{\mathcal{Z}(v,w,\boldsymbol{\theta})} - |\mathbb{E}_{u|v,w,\boldsymbol{\theta}}[u]|^2 \in \mathbb{R} \tag{3.13}$$

The expression for the conditional variance stems from

$$\mathrm{Var}_{u|v,w,\boldsymbol{\theta}}(u) := \frac{\int_u |u - \mathbb{E}_{u|v,w,\boldsymbol{\theta}}[u]|^2 p(u;\boldsymbol{\theta})\mathcal{N}(u;v,w)du}{\int_u p(u;\boldsymbol{\theta})\mathcal{N}(u;v,w)du} \tag{3.14}$$

$$N_2(v,w,\boldsymbol{\theta}) := \int_u |u - \mathbb{E}_{u|v,w,\boldsymbol{\theta}}[u]|^2 p(u;\boldsymbol{\theta})\mathcal{N}(u;v,w)du \tag{3.15}$$

$$\begin{aligned} &= \int_u |u|^2 p(u;\boldsymbol{\theta})\mathcal{N}(u;v,w)du \\ &\quad - \mathbb{E}_{u|v,w,\boldsymbol{\theta}}[u] \int_u u^* p(u;\boldsymbol{\theta})\mathcal{N}(u;v,w)du \\ &\qquad + |\mathbb{E}_{u|v,w,\boldsymbol{\theta}}[u]|^2 \int_u p(u;\boldsymbol{\theta})\mathcal{N}(u;v,w)du \\ &\qquad\quad - \mathbb{E}_{u|v,w,\boldsymbol{\theta}}[u]^* \int_u u p(u;\boldsymbol{\theta})\mathcal{N}(u;v,w)du \end{aligned} \tag{3.16}$$

$$\begin{aligned} &= \int_u |u|^2 p(u;\boldsymbol{\theta})\mathcal{N}(u;v,w)du \\ &\quad - \mathbb{E}_{u|v,w,\boldsymbol{\theta}}[u]\mathbb{E}_{u|v,w,\boldsymbol{\theta}}[u]^* \mathcal{Z}(v,w,\boldsymbol{\theta}) \\ &\qquad + |\mathbb{E}_{u|v,w,\boldsymbol{\theta}}[u]|^2 \mathcal{Z}(v,w,\boldsymbol{\theta}) \\ &\qquad\quad - \mathbb{E}_{u|v,w,\boldsymbol{\theta}}[u]^* \mathbb{E}_{u|v,w,\boldsymbol{\theta}}[u] \mathcal{Z}(v,w,\boldsymbol{\theta}) \end{aligned} \tag{3.17}$$

$$= \int_u |u|^2 p(u;\boldsymbol{\theta})\mathcal{N}(u;v,w)du - |\mathbb{E}_{u|v,w,\boldsymbol{\theta}}[u]|^2 \mathcal{Z}(v,w,\boldsymbol{\theta})^* \tag{3.18}$$

$$N_2(v,w,\boldsymbol{\theta}) = N_{2a}(v,w,\boldsymbol{\theta}) - N_{2b}(v,w,\boldsymbol{\theta}) \tag{3.19}$$

$$N_{2b}(v,w,\boldsymbol{\theta}) := |\mathbb{E}_{u|v,w,\boldsymbol{\theta}}[u]|^2 \mathcal{Z}(v,w,\boldsymbol{\theta})^* = |\mathbb{E}_{u|v,w,\boldsymbol{\theta}}[u]|^2 \mathcal{Z}(v,w,\boldsymbol{\theta}) \tag{3.20}$$

$$\mathrm{Var}_{u|v,w,\boldsymbol{\theta}}(u) = \frac{N_2(v,w,\boldsymbol{\theta})}{\mathcal{Z}(v,w,\boldsymbol{\theta})} \tag{3.21}$$

$$= \frac{N_{2a}(v,w,\boldsymbol{\theta}) - N_{2b}(v,w,\boldsymbol{\theta})}{\mathcal{Z}(v,w,\boldsymbol{\theta})} \tag{3.22}$$

$$= \frac{N_{a2}(v,w,\boldsymbol{\theta})}{\mathcal{Z}(v,w,\boldsymbol{\theta})} - |\mathbb{E}_{u|v,w,\boldsymbol{\theta}}[u]|^2 \frac{\mathcal{Z}(v,w,\boldsymbol{\theta})}{\mathcal{Z}(v,w,\boldsymbol{\theta})} \tag{3.23}$$

$$= \frac{N_{a2}(v,w,\boldsymbol{\theta})}{\mathcal{Z}(v,w,\boldsymbol{\theta})} - |\mathbb{E}_{u|v,w,\boldsymbol{\theta}}[u]|^2 \tag{3.24}$$

where * denotes complex conjugation.

3.3.1 Relation to Rangan's Channel Functions

Rangan's GAMP allows for obtaining both MAP and MMSE estimates depending on the choice of channel functions. For MAP estimates the channel functions are found from certain probability maximisation problems and are, thus, closely related to typical optimisation formulations used in e.g. sparse inference [1], [16]. For MMSE estimates the channel functions are expectations and variances as seen in the channel functions given in Sections 3.1 and 3.2. To allow for both types (MAP and MMSE) of channel functions, Rangan uses the differently defined channel functions $g_{in}, g'_{in}, g_{out}, g'_{out}$ [1], [16]. For the MMSE case, we have the following relations:

$$g_{in}(r,\boldsymbol{\theta}_I,s) = f_{\tilde{\alpha}}(s,r;\boldsymbol{\theta}_I) \tag{3.25}$$

$$g'_{in}(r,\boldsymbol{\theta}_I,s) = \frac{f_{\tilde{\alpha}}(s,r;\boldsymbol{\theta}_I)}{s} \tag{3.26}$$

$$g_{out}(o,y,v) = \frac{f_{\tilde{z}}(v,o;y,\boldsymbol{\theta}_o) - o}{v} = q \tag{3.27}$$

$$g'_{out}(o,y,v) = \frac{f_{\tilde{z}}(v,o;y,\boldsymbol{\theta}_o) - v}{v^2} = -u \tag{3.28}$$

3.4 General Sparse Input Channel

Sparsity is a typical *structure* assumed on $\boldsymbol{\alpha}$ in the reconstruction problem described in Section 1.2. Thus, when using GAMP to solve such reconstruction problems, one usually uses a sparsity promoting prior. For that reason, many of the input side channels presented in the literature are described by a probability density function which is a mixture of a Dirac delta function at zero and some other known proper density function [5], [28], i.e.

$$p(\alpha; \boldsymbol{\theta}_I) = (1-\tau)\delta_{\text{Dirac}}(\alpha) + \tau\varphi(\alpha; \boldsymbol{\theta}_I) \tag{3.29}$$

where $\tau \in [0; 1]$ is the signal density and $\varphi(\alpha; \boldsymbol{\theta}_I)$ is e.g. Gaussian, Laplace, Student's t or even itself a mixture density. Such a prior is sometimes also referred to as a spike-and-slab prior because of the (sparsity promoting) spike at zero and a slab part $\varphi(\alpha; \boldsymbol{\theta}_I)$. The spike part is typically referred to as the Bernoulli part as in e.g. a sparse Bernoulli-Gaussian prior [39]. In using the expression in Equation (3.29), it is important to realise that any manipulations are to be understood as being done "inside" an integral (such as an expectation) which makes the use of the Dirac delta function well defined.

Since the Dirac delta function integrates to 1 over the real line, it is easily seen that $p(\alpha; \boldsymbol{\theta}_I)$ in Equation (3.29) integrates to 1 for any proper probability density function $\varphi(\alpha; \boldsymbol{\theta}_I)$ and is, thus, a proper probability density itself. The GAMP approximated posterior in Equation (3.4) for the general sparse prior in Equation (3.29) is

$$\begin{aligned}
p(\alpha|\mathbf{y}; s, r, \boldsymbol{\theta}_I) &= \frac{((1-\tau)\delta_{\text{Dirac}}(\alpha) + \tau\varphi(\alpha; \boldsymbol{\theta}_I))\mathcal{N}(\alpha; r, s)}{\mathcal{Z}_I} && (3.30)\\
&= \frac{((1-\tau)\delta_{\text{Dirac}}(\alpha) + \tau\varphi(\alpha; \boldsymbol{\theta}_I))\mathcal{N}(\alpha; r, s)}{(1-\tau)\int_\alpha \mathcal{N}(\alpha; r, s)\delta_{\text{Dirac}}(\alpha)d\alpha + \tau\int_\alpha \varphi(\alpha; \boldsymbol{\theta}_I)\mathcal{N}(\alpha; r, s)d\alpha} && (3.31)\\
&= \frac{(1-\tau)\delta_{\text{Dirac}}(\alpha)\mathcal{N}(\alpha; r, s) + \tau\varphi(\alpha; \boldsymbol{\theta}_I)\mathcal{N}(\alpha; r, s)}{(1-\tau)\mathcal{Z}_\delta + \tau\mathcal{Z}_\varphi} && (3.32)\\
&= \frac{(1-\tau)\mathcal{N}(\alpha; r, s)\delta_{\text{Dirac}}(\alpha)}{(1-\tau)\mathcal{Z}_\delta + \tau\mathcal{Z}_\varphi} + \frac{\tau\mathcal{N}(\alpha; r, s)\varphi(\alpha; \boldsymbol{\theta}_I)}{(1-\tau)\mathcal{Z}_\delta + \tau\mathcal{Z}_\varphi} && (3.33)\\
&= \frac{(1-\tau)\mathcal{Z}_\delta}{(1-\tau)\mathcal{Z}_\delta + \tau\mathcal{Z}_\varphi}\frac{\mathcal{N}(\alpha; r, s)\delta_{\text{Dirac}}(\alpha)}{\mathcal{Z}_\delta}\\
&\quad + \frac{\tau\mathcal{Z}_\varphi}{(1-\tau)\mathcal{N}(0; r, s) + \tau\mathcal{Z}_\varphi}\frac{\mathcal{N}(\alpha; r, s)\varphi(\alpha; \boldsymbol{\theta}_I)}{\mathcal{Z}_\varphi} && (3.34)\\
&= \left(1 - \frac{\tau\mathcal{Z}_\varphi}{(1-\tau)\mathcal{Z}_\delta + \tau\mathcal{Z}_\varphi}\right)\frac{\mathcal{N}(\alpha; r, s)\delta_{\text{Dirac}}(\alpha)}{\mathcal{Z}_\delta}\\
&\quad + \frac{\tau\mathcal{Z}_\varphi}{(1-\tau)\mathcal{N}(0; r, s) + \tau\mathcal{Z}_\varphi}\frac{\mathcal{N}(\alpha; r, s)\varphi(\alpha; \boldsymbol{\theta}_I)}{\mathcal{Z}_\varphi} && (3.35)\\
&= (1-\pi(r, s, \boldsymbol{\theta}_I))\delta'_{\text{Dirac}}(\alpha) + \pi(r, s, \boldsymbol{\theta}_I)\varphi_{\alpha|\mathbf{y}; s, r, \boldsymbol{\theta}_I}(\alpha; \boldsymbol{\theta}_I) && (3.36)\\
&= (1-\pi(r, s, \boldsymbol{\theta}_I))\delta_{\text{Dirac}}(\alpha) + \pi(r, s, \boldsymbol{\theta}_I)\varphi_{\alpha|\mathbf{y}; s, r, \boldsymbol{\theta}_I}(\alpha; \boldsymbol{\theta}_I) && (3.37)
\end{aligned}$$

for

$$\mathcal{Z}_I := \int_\alpha ((1-\tau)\delta_{\text{Dirac}}(\alpha) + \tau\varphi(\alpha; \boldsymbol{\theta}_I))\mathcal{N}(\alpha; r, s)d\alpha \tag{3.38}$$

$$\mathcal{Z}_\delta := \int_\alpha \delta_{\text{Dirac}}(\alpha)\mathcal{N}(\alpha; r, s)d\alpha = \mathcal{N}(0; r, s) \tag{3.39}$$

$$\mathcal{Z}_\varphi := \int_\alpha \varphi(\alpha; \boldsymbol{\theta}_I)\mathcal{N}(\alpha; r, s)d\alpha \tag{3.40}$$

$$\pi(r, s, \boldsymbol{\theta}_I) := \frac{\tau\mathcal{Z}_\varphi}{(1-\tau)\mathcal{N}(0; r, s) + \tau\mathcal{Z}_\varphi} = \frac{1}{1 + \frac{(1-\tau)\mathcal{N}(0;r,s)}{\tau\mathcal{Z}_\varphi}} = \frac{1}{1 + \left(\frac{\tau\mathcal{Z}_\varphi}{(1-\tau)\mathcal{N}(0;r,s)}\right)^{-1}} \tag{3.41}$$

$$\varphi_{\alpha|\mathbf{y};s,r,\boldsymbol{\theta}_I}(\alpha;\boldsymbol{\theta}_I) := \frac{\mathcal{N}(\alpha;r,s)\varphi(\alpha;\boldsymbol{\theta}_I)}{\mathcal{Z}_\varphi} \tag{3.42}$$

$$\delta'_{\text{Dirac}}(\alpha) := \frac{\mathcal{N}(\alpha;r,s)\delta_{\text{Dirac}}(\alpha)}{\mathcal{Z}_\delta} \tag{3.43}$$

The equality of Equations (3.36) and (3.37) is based on the manipulations being done "inside" an integral. In this case, the sampling property of the Dirac delta function $\delta_{\text{Dirac}}(\alpha)$ "sifts" the value at $\alpha = 0$ which means that the $\delta'_{\text{Dirac}}(\alpha)$ function provides a scaling of $\mathcal{N}(0;r,s)$ which is cancelled by $\mathcal{Z}_\delta = \mathcal{N}(0;r,s)$ essentially reducing $\delta'_{\text{Dirac}}(\alpha)$ to $\delta_{\text{Dirac}}(\alpha)$.

Thus, as reported in [28], for the general sparse prior in Equation (3.29), we find that the GAMP posterior in Equation (3.37) is again a sparse density consisting of a Dirac delta at zero and the *GAMP posterior* $\varphi_{\alpha|\mathbf{y};s,r,\boldsymbol{\theta}_I}(\alpha;\boldsymbol{\theta}_I)$ of $\varphi(\alpha;\boldsymbol{\theta}_I)$ with a posteriori signal density (posterior support probabilities) $\pi(r,s,\boldsymbol{\theta}_I)$. As is seen from Equation (3.41), we have $\pi(r,s,\boldsymbol{\theta}_I) \in [0;1]$ as along as $\tau \in [0;1]$ and $\varphi(\alpha;\boldsymbol{\theta}_I)$ is a proper density since all quantities are non-negative. Thus, the GAMP approximated posterior $p(\alpha|\mathbf{y};s,r,\boldsymbol{\theta}_I)$ remains a proper density.

3.5 General Weighted Sparse Input Channel

If we assume that $\boldsymbol{\alpha}$ in the reconstruction problem in Section 1.2 is not only sparse but *structured sparse* in the sense that some of the coefficient values in $\boldsymbol{\alpha}$ are more likely to be sparse than others, we may consider an independent but non-identical general sparse weighted (GWS) input channel, i.e.

$$p(\alpha_j;\boldsymbol{\theta}_I) = (1 - w_j\tau)\delta_{\text{Dirac}}(\alpha_j) + w_j\tau\varphi(\alpha_j;[\boldsymbol{\theta}_I]_j) \tag{3.44}$$

where $\tau \in [0;1]$ models the overall signal density and the $w_j \in [0;1]$, $j = 1,\ldots,n$ are individual weights that model the belief about the sparsity of the individual coefficients. We note that the general weighted sparse input channel in Equation (3.44) reduces to the general sparse input channel in Equation (3.29) if $\forall j, w_j = 1$. Since the Dirac delta function integrates to 1 over the real line, it is easily seen that $p(\alpha_j;[\boldsymbol{\theta}_I)]_j$ in Equation (3.44) integrates to 1 for any proper probability density function $\varphi(\alpha_j;[\boldsymbol{\theta}_I]_j)$ and is, thus, a proper probability density itself.

Since the input channel acts independently on each element of $\boldsymbol{\alpha},\mathbf{s},\mathbf{r}$, everything still decouples in manipulations involving Equation (3.44). Thus, following the same path of derivations as was done in deriving Equation (3.29), we find that GAMP approximated posterior for the GWS prior in Equation (3.44) is

$$\begin{aligned} p(\alpha_j|\mathbf{y};s_j,r_j,[\boldsymbol{\theta}_I]_j) = (1 - \pi^{\text{w}}_j(r_j,s_j,[\boldsymbol{\theta}_I]_j))\delta_{\text{Dirac}}(\alpha_j) \\ + \pi^{\text{w}}_j(r_j,s_j,[\boldsymbol{\theta}_I]_j)\varphi_{\alpha_j|\mathbf{y};s_j,r_j,[\boldsymbol{\theta}_I]_j}(\alpha_j;[\boldsymbol{\theta}_I]_j) \end{aligned} \tag{3.45}$$

for

$$\pi^{\text{w}}_j(r_j,s_j,[\boldsymbol{\theta}_I]_j) := \frac{w_j\tau\mathcal{Z}_{\varphi_j}}{(1 - w_j\tau)\mathcal{N}(0;r_j,s_j) + w_j\tau\mathcal{Z}_{\varphi_j}} = \frac{w_j\tau\mathcal{Z}_{\varphi_j}}{\mathcal{Z}_{I_j}} \tag{3.46}$$

$$= \frac{1}{1 + \frac{(1-w_j\tau)\mathcal{N}(0;r_j,s_j)}{w_j\tau\mathcal{Z}_{\varphi_j}}} = \frac{1}{1 + \left(\frac{w_j\tau\mathcal{Z}_{\varphi_j}}{(1-w_j\tau)\mathcal{N}(0;r_j,s_j)}\right)^{-1}} \tag{3.47}$$

$$\mathcal{Z}_{I_j} := \int_{\alpha_j} ((1 - w_j\tau)\delta_{\text{Dirac}}(\alpha_j) + w_j\tau\varphi(\alpha_j;[\boldsymbol{\theta}_I]_j))\mathcal{N}(\alpha_j;r_j,s_j)d\alpha_j \tag{3.48}$$

$$\mathcal{Z}_{\varphi_j} := \int_{\alpha_j} \varphi(\alpha_j;[\boldsymbol{\theta}_I]_j)\mathcal{N}(\alpha_j;r_j,s_j)d\alpha_j \tag{3.49}$$

$$\varphi_{\alpha_j|\mathbf{y};s_j,r_j,[\boldsymbol{\theta}_I]_j}(\alpha_j;[\boldsymbol{\theta}_I]_j) := \frac{\mathcal{N}(\alpha_j;r_j,s_j)\varphi(\alpha_j;[\boldsymbol{\theta}_I]_j)}{\mathcal{Z}_{\varphi_j}} \tag{3.50}$$

Thus, for the GWS prior in Equation (3.44), the GAMP posterior in Equation (3.45) is again a sparse density with posterior signal densities $\pi^{\text{w}}_j(r_j,s_j,[\boldsymbol{\theta}_I]_j)$, $j = 1,\ldots,n$. Again, it is clear from Equation (3.46) that $\pi^{\text{w}}_j(r_j,s_j,[\boldsymbol{\theta}_I]_j) \in [0;1]$ as long as $w_j\tau \in [0;1]$ and $\varphi_{\alpha_j|\mathbf{y};s_j,r_j,[\boldsymbol{\theta}_I]_j}(\alpha_j;[\boldsymbol{\theta}_I]_j)$

is a proper density since all quantities in $\pi_j^{\mathrm{w}}(r_j, s_j, [\boldsymbol{\theta}_I]_j)$ are non-negative. Thus, the GAMP approximated posterior $p(\alpha_j|\mathbf{y}; s_j, r_j, [\boldsymbol{\theta}_I]_j)$ remains a proper density.

Now defining

$$N_{1\varphi_j} := \int_{\alpha_j} \alpha_j \varphi(\alpha_j; [\boldsymbol{\theta}_I]_j) \mathcal{N}(\alpha_j; r_j, s_j) d\alpha_j \tag{3.51}$$

$$N_{2a\varphi_j} := \int_{\alpha_j} |\alpha_j|^2 \varphi(\alpha_j; [\boldsymbol{\theta}_I]_j) \mathcal{N}(\alpha_j; r_j, s_j) d\alpha_j \tag{3.52}$$

$$\tag{3.53}$$

and using Equations (3.12), (3.13), and (3.45), we find that the MMSE GAMP input channel mean and variance functions are given by

$$f_{\bar{\alpha}_j}(s_j, r_j; [\boldsymbol{\theta}_I]_j) = \pi_j^{\mathrm{w}}(r_j, s_j, [\boldsymbol{\theta}_I]_j) \frac{N_{1\varphi_j}}{\mathcal{Z}_{\varphi_j}} \tag{3.54}$$

$$f_{\tilde{\alpha}_j}(s_j, r_j; [\boldsymbol{\theta}_I]_j) = \pi_j^{\mathrm{w}}(r_j, s_j, [\boldsymbol{\theta}_I]_j) \frac{N_{2a\varphi_j}}{\mathcal{Z}_{\varphi_j}} - |f_{\bar{\alpha}_j}(s_j, r_j; [\boldsymbol{\theta}_I]_j)|^2 \tag{3.55}$$

since the Dirac delta at zero does not contribute to the mean or the variance. Thus, the GWS input channel mean and variance functions may be expressed as scaled versions for the slab-part mean and variance functions with the scaling given by the posterior signal densities. This may be exploited in an implementation of the GWS input channel to separate the slab-part update from the GWS updates making it easy to re-use the GWS updates with different slab-part updates.

3.6 Analytic Expressions for Common Output Channels

In an implementation of the MMSE GAMP, one may in general have to resort to numerical integration to evaluate output the channel functions $f_{\bar{z}}$, $f_{\bar{z}}$. However, for some channels, it is possible to derive analytic solutions to the integrals involved in evaluating the channel functions. Here we present some output channels for which analytic solutions to the channel evaluation functions exist.

3.6.1 AWGN Output Channel

For an additive white Gaussian noise (AWGN) output channel with noise variance σ^2 ($\boldsymbol{\theta}_o = [\sigma^2]$), i.e.

$$p(y|z; \boldsymbol{\theta}_o) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-z)^2}{2\sigma^2}\right) \tag{3.56}$$

we have channel functions [1], [16] (Eqs. (41), (42), (43)):

$$f_{\bar{z}}(v, o; y, \boldsymbol{\theta}_o) = \frac{vy + \sigma^2 o}{\sigma^2 + v} \tag{3.57}$$

$$= o + \frac{v}{\sigma^2 + v}(y - o) \tag{3.58}$$

$$f_{\tilde{z}}(v, o; y, \boldsymbol{\theta}_o) = \frac{\sigma^2 v}{\sigma^2 + v} \tag{3.59}$$

$$= \frac{1}{\frac{1}{\sigma^2} + \frac{1}{v}} \tag{3.60}$$

Note that Equations (3.58) and (3.60) are the expressions suggested by Parker in [15]. However, in EM-BG/GM-GAMP [28] Equations (3.58) and (3.59) are mentioned. Mathematically, there is no difference in this choice. Numerically, however, there may be.

3.6.2 AWLN Output Channel

For an additive white Laplacian noise (AWLN) output channel with rate parameter $\lambda > 0$, ($\boldsymbol{\theta}_o = [\lambda]$), i.e.

$$p(y|z;\boldsymbol{\theta}_o) = \frac{\lambda}{2}\exp(-\lambda|y-z|) \tag{3.61}$$

we have channel functions [40] (Eqs. (22), (23)):

$$f_{\bar{z}}(v,o;y,\boldsymbol{\theta}_o) = y + \frac{\underline{\mathcal{Z}}_o}{\mathcal{Z}_o}\left(\underline{o} - \sqrt{v}\frac{\phi_{\mathcal{N}}\left(\frac{-\underline{o}}{\sqrt{v}}\right)}{\Phi_{\mathcal{N}}\left(\frac{-\underline{o}}{\sqrt{v}}\right)}\right) + \frac{\bar{\mathcal{Z}}_o}{\mathcal{Z}_o}\left(\bar{o} + \sqrt{v}\frac{\phi_{\mathcal{N}}\left(\frac{\bar{o}}{\sqrt{v}}\right)}{\Phi_{\mathcal{N}}\left(\frac{\bar{o}}{\sqrt{v}}\right)}\right) \tag{3.62}$$

$$\begin{aligned} f_{\tilde{z}}(v,o;y,\boldsymbol{\theta}_o) = &-(y^2 - f_{\bar{z}}(v,o;y,\boldsymbol{\theta}_o))^2 \\ &+ \frac{\underline{\mathcal{Z}}_o}{\mathcal{Z}_o}\left[v\left(1 - \frac{\phi_{\mathcal{N}}\left(\frac{-\underline{o}}{\sqrt{v}}\right)}{\Phi_{\mathcal{N}}\left(\frac{-\underline{o}}{\sqrt{v}}\right)}\left(\frac{\phi_{\mathcal{N}}\left(\frac{-\underline{o}}{\sqrt{v}}\right)}{\Phi_{\mathcal{N}}\left(\frac{-\underline{o}}{\sqrt{v}}\right)} - \frac{\underline{o}}{\sqrt{v}}\right)\right) + \left(\underline{o} - \sqrt{v}\frac{\phi_{\mathcal{N}}\left(\frac{-\underline{o}}{\sqrt{v}}\right)}{\Phi_{\mathcal{N}}\left(\frac{-\underline{o}}{\sqrt{v}}\right)}\right)^2\right] \\ &+ \frac{\bar{\mathcal{Z}}_o}{\mathcal{Z}_o}\left[v\left(1 - \frac{\phi_{\mathcal{N}}\left(\frac{\bar{o}}{\sqrt{v}}\right)}{\Phi_{\mathcal{N}}\left(\frac{\bar{o}}{\sqrt{v}}\right)}\left(\frac{\phi_{\mathcal{N}}\left(\frac{\bar{o}}{\sqrt{v}}\right)}{\Phi_{\mathcal{N}}\left(\frac{\bar{o}}{\sqrt{v}}\right)} + \frac{\bar{o}}{\sqrt{v}}\right)\right) + \left(\bar{o} + \sqrt{v}\frac{\phi_{\mathcal{N}}\left(\frac{\bar{o}}{\sqrt{v}}\right)}{\Phi_{\mathcal{N}}\left(\frac{\bar{o}}{\sqrt{v}}\right)}\right)^2\right] \end{aligned} \tag{3.63}$$

for

$$\mathcal{Z}_o = \underline{\mathcal{Z}}_o + \bar{\mathcal{Z}}_o \tag{3.64}$$

$$\underline{\mathcal{Z}}_o = \frac{\lambda}{2}\exp\left(\frac{1}{2}\lambda^2 v + \check{o}\lambda\right)\Phi_{\mathcal{N}}\left(\frac{-\underline{o}}{\sqrt{v}}\right) \tag{3.65}$$

$$\bar{\mathcal{Z}}_o = \frac{\lambda}{2}\exp\left(\frac{1}{2}\lambda^2 v - \check{o}\lambda\right)\Phi_{\mathcal{N}}\left(\frac{\bar{o}}{\sqrt{v}}\right) \tag{3.66}$$

$$\check{o} = o - y \tag{3.67}$$

$$\underline{o} = \check{o} + \lambda v \tag{3.68}$$

$$\bar{o} = \check{o} - \lambda v \tag{3.69}$$

$$\Phi_{\mathcal{N}}(\check{x}) = \int_{-\infty}^{\check{x}} \phi_{\mathcal{N}}(t)\,dt = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\check{x}} \exp\left(-\frac{t^2}{2}\right)dt \tag{3.70}$$

$$\phi_{\mathcal{N}}(\check{x}) = \mathcal{N}(\check{x},0,1) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{\check{x}^2}{2}\right) \tag{3.71}$$

The derivation of these channel functions is similar to the derivation of the channel functions for the i.i.d. Sparse Bernoulli-Laplace input channel in Section 3.7.1. Note that the scaled complementary error function may be used to achieve better numerical accuracy in an implementation as detailed in Section 3.7.1.1.

3.7 Analytic Expressions for Common Input Channels

In an implementation of the MMSE GAMP, one may in general have to resort to numerical integration to evaluate the input channel functions $f_{\bar{\alpha}}$, $f_{\tilde{\alpha}}$,. However, for some channels, it is possible to derive analytic solutions to the integrals involved in evaluating the channel functions. Here we present some input channels for which analytic solutions to the channel evaluation functions exist.

3.7.1 I.i.d. Sparse Bernoulli-Laplace Input Channel

We now consider an i.i.d. BL input channel with signal density τ, Laplace mean μ, and rate parameter $\lambda > 0$, ($\boldsymbol{\theta}_I = [\tau, \mu, \lambda]^T$), i.e. an input channel described by

$$p(\alpha;\boldsymbol{\theta}_I) = (1-\tau)\delta_{\text{Dirac}}(\alpha) + \tau\frac{\lambda}{2}\exp(-\lambda|\alpha-\mu|) \tag{3.72}$$

We derive the channel functions following the general procedure described in Section 3.3. Towards this end, we make use of various tricks and techniques used in the derivation of the elastic net prior in [41] as well as in the derivation of the ALWN output channel in [40]. Starting with the product of the prior and the Gaussian GAMP field, we observe that

$$p(\alpha;\boldsymbol{\theta}_I)\mathcal{N}(\alpha;r,s) = (1-\tau)\delta_{\text{Dirac}}(\alpha)\mathcal{N}(\alpha;r,s) + \tau\frac{\lambda}{2}\exp(-\lambda|\alpha-\mu|)\,\mathcal{N}(\alpha;r,s) \tag{3.73}$$

$$= (1-\tau)\delta_{\text{Dirac}}(\check{\alpha}+\mu)\mathcal{N}(\check{\alpha};\check{r},s) + \tau\frac{\lambda}{2}\exp(-\lambda|\check{\alpha}|)\,\mathcal{N}(\check{\alpha};\check{r},s) \tag{3.74}$$

where in Equation (3.74), we have shifted everything to align with the Laplace mean μ, i.e., $\check{\alpha} = \alpha - \mu$, $\check{r} = r - \mu$. Then the normalisation constant in Equation (3.9) is given by

$$\mathcal{Z}_I = (1-\tau)\int_{-\infty}^{\infty}\delta_{\text{Dirac}}(\check{\alpha}+\mu)\mathcal{N}(\check{\alpha};\check{r},s)d\check{\alpha} + \tau\int_{-\infty}^{\infty}\frac{\lambda}{2}\exp(-\lambda|\check{\alpha}|)\,\mathcal{N}(\check{\alpha};\check{r},s)d\check{\alpha} \tag{3.75}$$

$$= (1-\tau)\int_{-\infty}^{\infty}\delta_{\text{Dirac}}(\alpha)\mathcal{N}(\alpha;r,s)d\alpha + \tau\int_{-\infty}^{\infty}\frac{\lambda}{2}\exp(-\lambda|\check{\alpha}|)\,\mathcal{N}(\check{\alpha};\check{r},s)d\check{\alpha} \tag{3.76}$$

$$= (1-\tau)\mathcal{N}(0;r,s) + \tau\int_{-\infty}^{\infty}\frac{\lambda}{2}\exp(-\lambda|\check{\alpha}|)\,\mathcal{N}(\check{\alpha};\check{r},s)d\check{\alpha} \tag{3.77}$$

where we have used the sampling property of the generalised Dirac delta function. The absolute value in the integrand in Equation (3.77) requires considering the two cases: $\check{\alpha} < 0$, $\check{\alpha} > 0$, separately. For $\check{\alpha} < 0$, we have

$$\frac{\lambda}{2}\exp(\lambda\check{\alpha})\,\mathcal{N}(\check{\alpha};\check{r},s) = \frac{1}{\sqrt{2\pi s}}\frac{\lambda}{2}\exp\left(-\frac{(\check{\alpha}-\check{r})^2 - \lambda\check{\alpha}2s}{2s}\right) \tag{3.78}$$

$$= \frac{1}{\sqrt{2\pi s}}\frac{\lambda}{2}\exp\left(-\frac{\check{\alpha}^2+\check{r}^2-2\check{\alpha}(\check{r}+\lambda s)}{2s}\right) \tag{3.79}$$

$$= \frac{1}{\sqrt{2\pi s}}\frac{\lambda}{2}\exp\left(-\frac{(\check{\alpha}-(\check{r}+\lambda s))^2-(\lambda s)^2-2\check{r}\lambda s}{2s}\right) \tag{3.80}$$

$$= \frac{1}{\sqrt{2\pi s}}\frac{\lambda}{2}\exp\left(-\frac{(\check{\alpha}-\underline{r})^2}{2s}\right)\exp\left(\frac{1}{2}\lambda^2 s+\check{r}\lambda\right) \tag{3.81}$$

$$= \frac{\lambda}{2}\exp\left(\frac{1}{2}\lambda^2 s+\check{r}\lambda\right)\mathcal{N}(\check{\alpha};\underline{r},s) \tag{3.82}$$

for $\underline{r} = \check{r} + \lambda s$. Note that $\frac{1}{2}\lambda^2 s + \check{r}\lambda = \frac{\underline{r}^2-\check{r}^2}{2s}$. Similarly, for $\check{\alpha} > 0$, we have

$$\frac{\lambda}{2}\exp(-\lambda\check{\alpha})\,\mathcal{N}(\check{\alpha};\check{r},s) = \frac{1}{\sqrt{2\pi s}}\frac{\lambda}{2}\exp\left(-\frac{(\check{\alpha}-\check{r})^2 + \lambda\check{\alpha}2s}{2s}\right) \tag{3.83}$$

$$= \frac{1}{\sqrt{2\pi s}}\frac{\lambda}{2}\exp\left(-\frac{\check{\alpha}^2+\check{r}^2-2\check{\alpha}(\check{r}-\lambda s)}{2s}\right) \tag{3.84}$$

$$= \frac{1}{\sqrt{2\pi s}}\frac{\lambda}{2}\exp\left(-\frac{(\check{\alpha}-(\check{r}-\lambda s))^2-(\lambda s)^2+2\check{r}\lambda s}{2s}\right) \tag{3.85}$$

$$= \frac{1}{\sqrt{2\pi s}}\frac{\lambda}{2}\exp\left(-\frac{(\check{\alpha}-\bar{r})^2}{2s}\right)\exp\left(\frac{1}{2}\lambda^2 s-\check{r}\lambda\right) \tag{3.86}$$

$$= \frac{\lambda}{2}\exp\left(\frac{1}{2}\lambda^2 s-\check{r}\lambda\right)\mathcal{N}(\check{\alpha};\bar{r},s) \tag{3.87}$$

for $\bar{r} = \check{r} - \lambda s$. Note that $\frac{1}{2}\lambda^2 s - \check{r}\lambda = \frac{\bar{r}^2-\check{r}^2}{2s}$.

Now returning to the Equation (3.77), we may split the integral in a lower and an upper part

$$\begin{aligned}
\mathcal{Z}_I &= (1-\tau)\mathcal{N}(0;r,s) \\
&\quad + \tau \left(\int_{-\infty}^{0} \frac{\lambda}{2} \exp(\lambda \check{\alpha}) \mathcal{N}(\check{\alpha};\check{r},s) d\check{\alpha} + \int_{0}^{\infty} \frac{\lambda}{2} \exp(-\lambda \check{\alpha}) \mathcal{N}(\check{\alpha};\check{r},s) d\check{\alpha} \right) \qquad (3.88) \\
&= (1-\tau)\mathcal{N}(0;r,s) \\
&\quad + \tau \left(\frac{\lambda}{2} \exp\left(\frac{1}{2}\lambda^2 s + \check{r}\lambda \right) \int_{-\infty}^{0} \mathcal{N}(\check{\alpha};\underline{r},s) d\check{\alpha} \right. \\
&\qquad \left. + \frac{\lambda}{2} \exp\left(\frac{1}{2}\lambda^2 s - \check{r}\lambda \right) \int_{0}^{\infty} \mathcal{N}(\check{\alpha};\bar{r},s) d\check{\alpha} \right) \qquad (3.89) \\
&= (1-\tau)\mathcal{N}(0;r,s) \\
&\quad + \tau \left(\frac{\lambda}{2} \exp\left(\frac{1}{2}\lambda^2 s + \check{r}\lambda \right) \Phi_{\mathcal{N}}\left(\frac{-\underline{r}}{\sqrt{s}} \right) + \frac{\lambda}{2} \exp\left(\frac{1}{2}\lambda^2 s - \check{r}\lambda \right) \left(1 - \Phi_{\mathcal{N}}\left(\frac{-\bar{r}}{\sqrt{s}} \right) \right) \right) \qquad (3.90) \\
&= (1-\tau)\mathcal{N}(0;r,s) \\
&\quad + \tau \left(\frac{\lambda}{2} \exp\left(\frac{1}{2}\lambda^2 s + \check{r}\lambda \right) \Phi_{\mathcal{N}}\left(\frac{-\underline{r}}{\sqrt{s}} \right) + \frac{\lambda}{2} \exp\left(\frac{1}{2}\lambda^2 s - \check{r}\lambda \right) \Phi_{\mathcal{N}}\left(\frac{\bar{r}}{\sqrt{s}} \right) \right) \qquad (3.91) \\
&= (1-\tau)\mathcal{N}(0;r,s) + \tau(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I) \qquad (3.92)
\end{aligned}$$

for

$$\begin{aligned}
\underline{\mathcal{Z}}_I &= \frac{\lambda}{2} \exp\left(\frac{1}{2}\lambda^2 s + \check{r}\lambda \right) \Phi_{\mathcal{N}}\left(\frac{-\underline{r}}{\sqrt{s}} \right) \qquad (3.93) \\
\bar{\mathcal{Z}}_I &= \frac{\lambda}{2} \exp\left(\frac{1}{2}\lambda^2 s - \check{r}\lambda \right) \Phi_{\mathcal{N}}\left(\frac{\bar{r}}{\sqrt{s}} \right) \qquad (3.94)
\end{aligned}$$

where we have introduced the cumulative distribution function (cdf) $\Phi_{\mathcal{N}}(\check{x}) = \int_{-\infty}^{\check{x}} \phi_{\mathcal{N}}(t)\, dt = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\check{x}} \exp\left(-\frac{t^2}{2}\right) dt$ of a standard normal distribution (with probability density function (pdf) $\phi_{\mathcal{N}}(\check{x}) = \mathcal{N}(\check{x}, 0, 1) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\check{x}^2}{2}\right)$). Note that we have the following symmetry relations

$$\begin{aligned}
\phi_{\mathcal{N}}(-\check{x}) &= \phi_{\mathcal{N}}(\check{x}) \qquad (3.95) \\
\Phi_{\mathcal{N}}(-\check{x}) &= 1 - \Phi_{\mathcal{N}}(\check{x}) \qquad (3.96)
\end{aligned}$$

Using techniques similar to those used above for deriving $\mathcal{Z}_I$, we have the following expression for the N_1 quantity in Equation (3.10)

$$\begin{aligned}
N_1 &= (1-\tau) \int_{-\infty}^{\infty} (\check{\alpha} + \mu) \delta_{\text{Dirac}}(\check{\alpha} + \mu) \mathcal{N}(\check{\alpha};\check{r},s) d\check{\alpha} \\
&\quad + \tau \int_{-\infty}^{\infty} (\check{\alpha} + \mu) \frac{\lambda}{2} \exp(-\lambda |\check{\alpha}|) \mathcal{N}(\check{\alpha};\check{r},s) d\check{\alpha} \qquad (3.97) \\
&= (1-\tau) \int_{-\infty}^{\infty} \alpha \delta_{\text{Dirac}}(\alpha) \mathcal{N}(\alpha;r,s) d\alpha \\
&\quad + \tau \int_{-\infty}^{\infty} (\check{\alpha} + \mu) \frac{\lambda}{2} \exp(-\lambda |\check{\alpha}|) \mathcal{N}(\check{\alpha};\check{r},s) d\check{\alpha} \qquad (3.98) \\
&= \tau \int_{-\infty}^{\infty} (\check{\alpha} + \mu) \frac{\lambda}{2} \exp(-\lambda |\check{\alpha}|) \mathcal{N}(\check{\alpha};\check{r},s) d\check{\alpha} \qquad (3.99) \\
&= \mu\tau \int_{-\infty}^{\infty} \frac{\lambda}{2} \exp(-\lambda |\check{\alpha}|) \mathcal{N}(\check{\alpha};\check{r},s) d\check{\alpha} + \tau \int_{-\infty}^{\infty} \check{\alpha} \frac{\lambda}{2} \exp(-\lambda |\check{\alpha}|) \mathcal{N}(\check{\alpha};\check{r},s) d\check{\alpha} \qquad (3.100) \\
&= \mu\tau(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I) + \tau \int_{-\infty}^{\infty} \check{\alpha} \frac{\lambda}{2} \exp(-\lambda |\check{\alpha}|) \mathcal{N}(\check{\alpha};\check{r},s) d\check{\alpha} \qquad (3.101)
\end{aligned}$$

$$\begin{aligned}
&= \tau\mu(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I) \\
&\quad + \tau\left(\int_{-\infty}^{0} \check{\alpha}\frac{\lambda}{2}\exp(\lambda\check{\alpha})\,\mathcal{N}(\check{\alpha};\check{r},s)d\check{\alpha} + \int_{0}^{\infty} \check{\alpha}\frac{\lambda}{2}\exp(-\lambda\check{\alpha})\,\mathcal{N}(\check{\alpha};\check{r},s)d\check{\alpha}\right) && (3.102) \\
&= \tau\mu(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I) + \tau\left(\frac{\lambda}{2}\exp\!\left(\frac{1}{2}\lambda^2 s + \check{r}\lambda\right)\int_{-\infty}^{0} \check{\alpha}\mathcal{N}(\check{\alpha};\underline{r},s)d\check{\alpha}\right. \\
&\quad \left. + \frac{\lambda}{2}\exp\!\left(\frac{1}{2}\lambda^2 s - \check{r}\lambda\right)\int_{0}^{\infty} \check{\alpha}\mathcal{N}(\check{\alpha};\bar{r},s)d\check{\alpha}\right) && (3.103) \\
&= \tau\mu(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I) \\
&\quad + \tau\left(\frac{\lambda}{2}\exp\!\left(\frac{1}{2}\lambda^2 s + \check{r}\lambda\right)\Phi_{\mathcal{N}}\!\left(\frac{-\underline{r}}{\sqrt{s}}\right)\int_{-\infty}^{0} \check{\alpha}\frac{\mathcal{N}(\check{\alpha};\underline{r},s)}{\Phi_{\mathcal{N}}\left(\frac{-\underline{r}}{\sqrt{s}}\right)}d\check{\alpha}\right. \\
&\qquad \left. + \frac{\lambda}{2}\exp\!\left(\frac{1}{2}\lambda^2 s - \check{r}\lambda\right)\Phi_{\mathcal{N}}\!\left(\frac{\bar{r}}{\sqrt{s}}\right)\int_{0}^{\infty} \check{\alpha}\frac{\mathcal{N}(\check{\alpha};\bar{r},s)}{\Phi_{\mathcal{N}}\left(\frac{\bar{r}}{\sqrt{s}}\right)}d\check{\alpha}\right) && (3.104) \\
&= \tau\mu(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I) + \tau\left(\underline{\mathcal{Z}}_I\int_{-\infty}^{0} \check{\alpha}\frac{\frac{1}{\sqrt{s}}\phi_{\mathcal{N}}\left(\frac{\check{\alpha}-\underline{r}}{\sqrt{s}}\right)}{\Phi_{\mathcal{N}}\left(\frac{-\underline{r}}{\sqrt{s}}\right)}d\check{\alpha} + \bar{\mathcal{Z}}_I\int_{0}^{\infty} \check{\alpha}\frac{\frac{1}{\sqrt{s}}\phi_{\mathcal{N}}\left(\frac{\check{\alpha}-\bar{r}}{\sqrt{s}}\right)}{\Phi_{\mathcal{N}}\left(\frac{\bar{r}}{\sqrt{s}}\right)}d\check{\alpha}\right) && (3.105) \\
&= \tau\mu(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I) + \tau\left(\underline{\mathcal{Z}}_I\int_{-\infty}^{0} \check{\alpha}\frac{\frac{1}{\sqrt{s}}\phi_{\mathcal{N}}\left(\frac{\check{\alpha}-\underline{r}}{\sqrt{s}}\right)}{\Phi_{\mathcal{N}}\left(\frac{-\underline{r}}{\sqrt{s}}\right)}d\check{\alpha} + \bar{\mathcal{Z}}_I\int_{0}^{\infty} \check{\alpha}\frac{\frac{1}{\sqrt{s}}\phi_{\mathcal{N}}\left(\frac{\check{\alpha}-\bar{r}}{\sqrt{s}}\right)}{1-\Phi_{\mathcal{N}}\left(\frac{-\bar{r}}{\sqrt{s}}\right)}d\check{\alpha}\right) && (3.106)
\end{aligned}$$

Now, consider the *double truncated normal distribution* with probability density function [42]

$$\mathcal{TN}(\check{x},\xi,\sigma^2,a,b) = \begin{cases} 0, & \check{x} < a \\ \frac{\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(\check{x}-\xi)^2}{2\sigma^2}\right)}{\int_a^b \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(t-\xi)^2}{2\sigma^2}\right)dt} = \frac{\phi_{\mathcal{N}}\left(\frac{\check{x}-\xi}{\sigma}\right)}{\sigma\left(\Phi_{\mathcal{N}}\left(\frac{b-\xi}{\sigma}\right)-\Phi_{\mathcal{N}}\left(\frac{a-\xi}{\sigma}\right)\right)}, & a \le \check{x} \le b \\ 0, & \check{x} > b \end{cases} \tag{3.107}$$

with mean and variance [42]

$$\begin{aligned}
\mathbb{E}[\check{x}] &= \int_{-\infty}^{\infty} \check{x}\mathcal{TN}(\check{x},\xi,\sigma^2,a,b)d\check{x} && (3.108) \\
&= \xi + \sigma\frac{\phi_{\mathcal{N}}\left(\frac{a-\xi}{\sigma}\right) - \phi_{\mathcal{N}}\left(\frac{b-\xi}{\sigma}\right)}{\Phi_{\mathcal{N}}\left(\frac{b-\xi}{\sigma}\right) - \Phi_{\mathcal{N}}\left(\frac{a-\xi}{\sigma}\right)} && (3.109) \\
\mathrm{Var}(\check{x}) &= \int_{-\infty}^{\infty} (\check{x}-\mathbb{E}[\check{x}])^2\mathcal{TN}(\check{x},\xi,\sigma^2,a,b)d\check{x} && (3.110) \\
&= \sigma^2\left(1 + \frac{\frac{a-\xi}{\sigma}\phi_{\mathcal{N}}\left(\frac{a-\xi}{\sigma}\right) - \frac{b-\xi}{\sigma}\phi_{\mathcal{N}}\left(\frac{b-\xi}{\sigma}\right)}{\Phi_{\mathcal{N}}\left(\frac{b-\xi}{\sigma}\right) - \Phi_{\mathcal{N}}\left(\frac{a-\xi}{\sigma}\right)} - \left(\frac{\phi_{\mathcal{N}}\left(\frac{a-\xi}{\sigma}\right) - \phi_{\mathcal{N}}\left(\frac{b-\xi}{\sigma}\right)}{\Phi_{\mathcal{N}}\left(\frac{b-\xi}{\sigma}\right) - \Phi_{\mathcal{N}}\left(\frac{a-\xi}{\sigma}\right)}\right)^2\right) && (3.111)
\end{aligned}$$

Returning to Equation (3.106), we find that the integrals in that equation correspond to the mean values of two *singly truncated normal distributions*

$$N_1 = \tau\mu(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I) + \tau\left(\underline{\mathcal{Z}}_I\int_{-\infty}^{0} \check{\alpha}\mathcal{TN}(\check{\alpha},\underline{r},s,-\infty,0)d\check{\alpha} + \bar{\mathcal{Z}}_I\int_{0}^{\infty} \check{\alpha}\mathcal{TN}(\check{\alpha},\bar{r},s,0,\infty)d\check{\alpha}\right) \tag{3.112}$$

Thus, using the expression in Equation (3.109) (handling $\phi_{\mathcal{N}}(\infty)$ properly as a limit - see also [43]), we have

$$N_1 = \tau\mu(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I) + \tau\left(\underline{\mathcal{Z}}_I\left(\underline{r} + \sqrt{s}\frac{-\phi_{\mathcal{N}}\left(\frac{-\underline{r}}{\sqrt{s}}\right)}{\Phi_{\mathcal{N}}\left(\frac{-\underline{r}}{\sqrt{s}}\right)}\right) + \bar{\mathcal{Z}}_I\left(\bar{r} + \sqrt{s}\frac{\phi_{\mathcal{N}}\left(\frac{-\bar{r}}{\sqrt{s}}\right)}{1 - \Phi_{\mathcal{N}}\left(\frac{-\bar{r}}{\sqrt{s}}\right)}\right)\right) \tag{3.113}$$

$$= \tau\mu(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I) + \tau\left(\underline{\mathcal{Z}}_I\left(\underline{r} - \sqrt{s}\frac{\phi_{\mathcal{N}}\left(\frac{-\underline{r}}{\sqrt{s}}\right)}{\Phi_{\mathcal{N}}\left(\frac{-\underline{r}}{\sqrt{s}}\right)}\right) + \bar{\mathcal{Z}}_I\left(\bar{r} + \sqrt{s}\frac{\phi_{\mathcal{N}}\left(\frac{\bar{r}}{\sqrt{s}}\right)}{\Phi_{\mathcal{N}}\left(\frac{\bar{r}}{\sqrt{s}}\right)}\right)\right) \tag{3.114}$$

Using techniques similar to those used above for deriving N_1, we have the following expression for the N_{2a} quantity in Equation (3.11)

$$\begin{aligned}
N_{2a} &= (1-\tau)\int_{-\infty}^{\infty}(\check{\alpha}+\mu)^2\delta_{\text{Dirac}}(\check{\alpha}+\mu)\mathcal{N}(\check{\alpha};\check{r},s)d\check{\alpha} \\
&\quad + \tau\int_{-\infty}^{\infty}(\check{\alpha}+\mu)^2\frac{\lambda}{2}\exp(-\lambda|\check{\alpha}|)\,\mathcal{N}(\check{\alpha};\check{r},s)d\check{\alpha} && (3.115)\\
&= (1-\tau)\int_{-\infty}^{\infty}\alpha^2\delta_{\text{Dirac}}(\alpha)\mathcal{N}(\alpha;r,s)d\alpha \\
&\quad + \tau\int_{-\infty}^{\infty}(\check{\alpha}+\mu)^2\frac{\lambda}{2}\exp(-\lambda|\check{\alpha}|)\,\mathcal{N}(\check{\alpha};\check{r},s)d\check{\alpha} && (3.116)\\
&= \tau\int_{-\infty}^{\infty}(\check{\alpha}+\mu)^2\frac{\lambda}{2}\exp(-\lambda|\check{\alpha}|)\,\mathcal{N}(\check{\alpha};\check{r},s)d\check{\alpha} && (3.117)\\
&= \tau\int_{-\infty}^{\infty}(\mu^2+2\mu\check{\alpha}+\check{\alpha}^2)\frac{\lambda}{2}\exp(-\lambda|\check{\alpha}|)\,\mathcal{N}(\check{\alpha};\check{r},s)d\check{\alpha} && (3.118)\\
&= \tau\mu^2(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I) + 2\mu(N_1 - \tau\mu(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I)) \\
&\quad + \tau\int_{-\infty}^{\infty}\check{\alpha}^2\frac{\lambda}{2}\exp(-\lambda|\check{\alpha}|)\,\mathcal{N}(\check{\alpha};\check{r},s)d\check{\alpha} && (3.119)\\
&= -\tau\mu^2(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I) + 2\mu N_1 \\
&\quad + \tau\left(\int_{-\infty}^{0}\check{\alpha}^2\frac{\lambda}{2}\exp(\lambda\check{\alpha})\,\mathcal{N}(\check{\alpha};\check{r},s)d\check{\alpha} + \int_{0}^{\infty}\check{\alpha}^2\frac{\lambda}{2}\exp(-\lambda\check{\alpha})\,\mathcal{N}(\check{\alpha};\check{r},s)d\check{\alpha}\right) && (3.120)\\
&= -\tau\mu^2(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I) + 2\mu N_1 \\
&\quad + \tau\left(\underline{\mathcal{Z}}_I\int_{-\infty}^{0}\check{\alpha}^2\frac{\frac{1}{\sqrt{s}}\phi_{\mathcal{N}}\left(\frac{\check{\alpha}-\underline{r}}{\sqrt{s}}\right)}{\Phi_{\mathcal{N}}\left(\frac{-\underline{r}}{\sqrt{s}}\right)}d\check{\alpha} + \bar{\mathcal{Z}}_I\int_{0}^{\infty}\check{\alpha}^2\frac{\frac{1}{\sqrt{s}}\phi_{\mathcal{N}}\left(\frac{\check{\alpha}-\bar{r}}{\sqrt{s}}\right)}{1-\Phi_{\mathcal{N}}\left(\frac{-\bar{r}}{\sqrt{s}}\right)}d\check{\alpha}\right) && (3.121)
\end{aligned}$$

The integrals in Equation (3.121) correspond to the second moments of two *singly truncated normal distributions*

$$\begin{aligned}
N_{2a} &= -\tau\mu^2(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I) + 2\mu N_1 \\
&\quad + \tau\left(\underline{\mathcal{Z}}_I\int_{-\infty}^{0}\check{\alpha}^2\mathcal{TN}(\check{\alpha},\underline{r},s,-\infty,0)d\check{\alpha} + \bar{\mathcal{Z}}_I\int_{0}^{\infty}\check{\alpha}^2\mathcal{TN}(\check{\alpha},\bar{r},s,0,\infty)d\check{\alpha}\right) && (3.122)
\end{aligned}$$

Thus, using $\mathbb{E}[\check{x}^2] = \mathrm{Var}(\check{x}) + \mathbb{E}[\check{x}]^2$ together with Equation (3.111), we get

$$\begin{aligned}
N_{2a} &= -\tau\mu^2(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I) + 2\mu N_1 \\
&\quad + \tau \left(\underline{\mathcal{Z}}_I \left[s \left(1 + \frac{-\frac{-\underline{r}}{\sqrt{s}}\phi_\mathcal{N}\left(\frac{-\underline{r}}{\sqrt{s}}\right)}{\Phi_\mathcal{N}\left(\frac{-\underline{r}}{\sqrt{s}}\right)} - \left(\frac{-\phi_\mathcal{N}\left(\frac{-\underline{r}}{\sqrt{s}}\right)}{\Phi_\mathcal{N}\left(\frac{-\underline{r}}{\sqrt{s}}\right)} \right)^2 \right) + \left(\underline{r} - \sqrt{s}\frac{\phi_\mathcal{N}\left(\frac{-\underline{r}}{\sqrt{s}}\right)}{\Phi_\mathcal{N}\left(\frac{-\underline{r}}{\sqrt{s}}\right)} \right)^2 \right] \right. \\
&\quad \left. + \bar{\mathcal{Z}}_I \left[s \left(1 + \frac{\frac{-\bar{r}}{\sqrt{s}}\phi_\mathcal{N}\left(\frac{-\bar{r}}{\sqrt{s}}\right)}{1 - \Phi_\mathcal{N}\left(\frac{-\bar{r}}{\sqrt{s}}\right)} - \left(\frac{\phi_\mathcal{N}\left(\frac{-\bar{r}}{\sqrt{s}}\right)}{1 - \Phi_\mathcal{N}\left(\frac{-\bar{r}}{\sqrt{s}}\right)} \right)^2 \right) + \left(\bar{r} + \sqrt{s}\frac{\phi_\mathcal{N}\left(\frac{\bar{r}}{\sqrt{s}}\right)}{\Phi_\mathcal{N}\left(\frac{\bar{r}}{\sqrt{s}}\right)} \right)^2 \right] \right) \qquad (3.123) \\
&= 2\mu N_1 - \tau\mu^2(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I) \\
&\quad + \tau \left(\underline{\mathcal{Z}}_I \left[s \left(1 - \frac{\phi_\mathcal{N}\left(\frac{-\underline{r}}{\sqrt{s}}\right)}{\Phi_\mathcal{N}\left(\frac{-\underline{r}}{\sqrt{s}}\right)} \left(\frac{\phi_\mathcal{N}\left(\frac{-\underline{r}}{\sqrt{s}}\right)}{\Phi_\mathcal{N}\left(\frac{-\underline{r}}{\sqrt{s}}\right)} - \frac{\underline{r}}{\sqrt{s}} \right) \right) + \left(\underline{r} - \sqrt{s}\frac{\phi_\mathcal{N}\left(\frac{-\underline{r}}{\sqrt{s}}\right)}{\Phi_\mathcal{N}\left(\frac{-\underline{r}}{\sqrt{s}}\right)} \right)^2 \right] \right. \\
&\quad \left. + \bar{\mathcal{Z}}_I \left[s \left(1 - \frac{\phi_\mathcal{N}\left(\frac{\bar{r}}{\sqrt{s}}\right)}{\Phi_\mathcal{N}\left(\frac{\bar{r}}{\sqrt{s}}\right)} \left(\frac{\phi_\mathcal{N}\left(\frac{\bar{r}}{\sqrt{s}}\right)}{\Phi_\mathcal{N}\left(\frac{\bar{r}}{\sqrt{s}}\right)} + \frac{\bar{r}}{\sqrt{s}} \right) \right) + \left(\bar{r} + \sqrt{s}\frac{\phi_\mathcal{N}\left(\frac{\bar{r}}{\sqrt{s}}\right)}{\Phi_\mathcal{N}\left(\frac{\bar{r}}{\sqrt{s}}\right)} \right)^2 \right] \right) \qquad (3.124)
\end{aligned}$$

Finally, we arrive at the following expressions for the i.i.d. BL channel functions based on Equations (3.12) and (3.13)

$$\begin{aligned}
f_{\bar{\alpha}}(s, r; \boldsymbol{\theta}_I) &= \frac{N_1}{\mathcal{Z}_I} \qquad (3.125) \\
&= \frac{\tau\mu(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I) + \tau\left(\underline{\mathcal{Z}}_I \left(\underline{r} - \sqrt{s}\frac{\phi_\mathcal{N}\left(\frac{-\underline{r}}{\sqrt{s}}\right)}{\Phi_\mathcal{N}\left(\frac{-\underline{r}}{\sqrt{s}}\right)} \right) + \bar{\mathcal{Z}}_I \left(\bar{r} + \sqrt{s}\frac{\phi_\mathcal{N}\left(\frac{\bar{r}}{\sqrt{s}}\right)}{\Phi_\mathcal{N}\left(\frac{\bar{r}}{\sqrt{s}}\right)} \right) \right)}{\mathcal{Z}_I} \qquad (3.126) \\
&= \tau\left(\mu\frac{(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I)}{\mathcal{Z}_I} + \frac{\underline{\mathcal{Z}}_I}{\mathcal{Z}_I}\left(\underline{r} - \sqrt{s}\frac{\phi_\mathcal{N}\left(\frac{-\underline{r}}{\sqrt{s}}\right)}{\Phi_\mathcal{N}\left(\frac{-\underline{r}}{\sqrt{s}}\right)} \right) + \frac{\bar{\mathcal{Z}}_I}{\mathcal{Z}_I}\left(\bar{r} + \sqrt{s}\frac{\phi_\mathcal{N}\left(\frac{\bar{r}}{\sqrt{s}}\right)}{\Phi_\mathcal{N}\left(\frac{\bar{r}}{\sqrt{s}}\right)} \right) \right) \qquad (3.127) \\
f_{\tilde{\alpha}}(s, r; \boldsymbol{\theta}_I) &= \frac{N_{2a}}{\mathcal{Z}_I} - f_{\bar{\alpha}}(s, r; \boldsymbol{\theta}_I)^2 \qquad (3.128) \\
&= 2\mu f_{\bar{\alpha}}(s, r; \boldsymbol{\theta}_I) + \tau\left\{ -\mu^2\frac{(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I)}{\mathcal{Z}_I} \right. \\
&\quad + \frac{\underline{\mathcal{Z}}_I}{\mathcal{Z}_\mathcal{I}}\left[s\left(1 - \frac{\phi_\mathcal{N}\left(\frac{-\underline{r}}{\sqrt{s}}\right)}{\Phi_\mathcal{N}\left(\frac{-\underline{r}}{\sqrt{s}}\right)} \left(\frac{\phi_\mathcal{N}\left(\frac{-\underline{r}}{\sqrt{s}}\right)}{\Phi_\mathcal{N}\left(\frac{-\underline{r}}{\sqrt{s}}\right)} - \frac{\underline{r}}{\sqrt{s}} \right) \right) + \left(\underline{r} - \sqrt{s}\frac{\phi_\mathcal{N}\left(\frac{-\underline{r}}{\sqrt{s}}\right)}{\Phi_\mathcal{N}\left(\frac{-\underline{r}}{\sqrt{s}}\right)} \right)^2 \right] \\
&\quad \left. + \frac{\bar{\mathcal{Z}}_I}{\mathcal{Z}_I}\left[s\left(1 - \frac{\phi_\mathcal{N}\left(\frac{\bar{r}}{\sqrt{s}}\right)}{\Phi_\mathcal{N}\left(\frac{\bar{r}}{\sqrt{s}}\right)} \left(\frac{\phi_\mathcal{N}\left(\frac{\bar{r}}{\sqrt{s}}\right)}{\Phi_\mathcal{N}\left(\frac{\bar{r}}{\sqrt{s}}\right)} + \frac{\bar{r}}{\sqrt{s}} \right) \right) + \left(\bar{r} + \sqrt{s}\frac{\phi_\mathcal{N}\left(\frac{\bar{r}}{\sqrt{s}}\right)}{\Phi_\mathcal{N}\left(\frac{\bar{r}}{\sqrt{s}}\right)} \right)^2 \right] \right\} \\
&\quad - f_{\bar{\alpha}}(s, r; \boldsymbol{\theta}_I)^2 \qquad (3.129)
\end{aligned}$$

for

$$\begin{aligned}
\mathcal{Z}_I &= (1-\tau)\mathcal{N}(0; r, s) + \tau(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I) \qquad (3.130) \\
\underline{\mathcal{Z}}_I &= \frac{\lambda}{2}\exp\left(\frac{1}{2}\lambda^2 s + \check{r}\lambda \right) \Phi_\mathcal{N}\left(\frac{-\underline{r}}{\sqrt{s}} \right) \qquad (3.131)
\end{aligned}$$

$$\bar{\mathcal{Z}}_I = \frac{\lambda}{2}\exp\left(\frac{1}{2}\lambda^2 s - \check{r}\lambda\right)\Phi_{\mathcal{N}}\left(\frac{\bar{r}}{\sqrt{s}}\right) \tag{3.132}$$

$$\check{r} = r - \mu \tag{3.133}$$

$$\underline{r} = \check{r} + \lambda s \tag{3.134}$$

$$\bar{r} = \check{r} - \lambda s \tag{3.135}$$

3.7.1.1 Numerical Accuracy Considerations for the i.i.d. BL Input Channel

When implementing these channel functions, one has to pay attention to fractions of the type $\frac{\phi_{\mathcal{N}}(\check{x})}{\Phi_{\mathcal{N}}(\check{x})}$. For such fractions, improved numerical accuracy may be obtained[1] by using a reasonable implementation[2] of the scaled complementary error function [44]

$$\operatorname{erfcx}(\check{x}) \coloneqq \exp\left(\check{x}^2\right)\frac{2}{\sqrt{\pi}}\int_{\check{x}}^{\infty}\exp\left(-t^2\right)dt \tag{3.136}$$

For the scaled complementary error function, we have

$$\begin{aligned}
\operatorname{erfcx}\left(\frac{-\check{x}}{\sqrt{2}}\right) &= \exp\left(\left(\frac{-\check{x}}{\sqrt{2}}\right)^2\right)\frac{2}{\sqrt{\pi}}\int_{\frac{-\check{x}}{\sqrt{2}}}^{\infty}\exp\left(-t^2\right)dt && (3.137)\\
&= \exp\left(\frac{\check{x}^2}{2}\right)\frac{2}{\sqrt{\pi}}\int_{-\infty}^{\frac{\check{x}}{\sqrt{2}}}\exp\left(-t^2\right)dt && (3.138)\\
&= \exp\left(\frac{\check{x}^2}{2}\right)\frac{2}{\sqrt{2\pi}}\int_{-\infty}^{\check{x}}\exp\left(-\left(\frac{t}{\sqrt{2}}\right)^2\right)dt && (3.139)\\
&= \exp\left(\frac{\check{x}^2}{2}\right)\frac{2}{\sqrt{2\pi}}\int_{-\infty}^{\check{x}}\exp\left(-\frac{t^2}{2}\right)dt && (3.140)\\
&= \exp\left(\frac{\check{x}^2}{2}\right)2\Phi_{\mathcal{N}}(\check{x}) && (3.141)
\end{aligned}$$

which means that

$$\begin{aligned}
\frac{\phi_{\mathcal{N}}(\check{x})}{\Phi_{\mathcal{N}}(\check{x})} &= \frac{\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{\check{x}^2}{2}\right)}{\frac{\operatorname{erfcx}\left(\frac{-\check{x}}{\sqrt{2}}\right)}{2\exp\left(\frac{\check{x}^2}{2}\right)}} && (3.142)\\
&= \frac{\frac{2}{\sqrt{2\pi}}\exp\left(-\frac{\check{x}^2}{2}\right)\exp\left(\frac{\check{x}^2}{2}\right)}{\operatorname{erfcx}\left(\frac{-\check{x}}{\sqrt{2}}\right)} && (3.143)\\
&= \frac{\frac{2}{\sqrt{2\pi}}}{\operatorname{erfcx}\left(\frac{-\check{x}}{\sqrt{2}}\right)} && (3.144)
\end{aligned}$$

[1] This use of complementary error function was inspired by its use in the `ElasticNetEstimIn.m` file in the GAMPMatlab Toolbox version 20161005 available at `https://sourceforge.net/projects/gampmatlab/`. See also Section 7.2 for more information about the GAMPMatlab Toolbox.

[2] See: `http://scipy.github.io/devdocs/special.html#error-function-and-fresnel-integrals` and `http://ab-initio.mit.edu/wiki/index.php/Faddeeva_Package`

3.7.1.2 BL Expressions for use with the GWS Input Channel

Based on the result in Equations (3.127) and (3.129), we may identify the following i.i.d BL channel updates to be used in the GWS framework described in Section 3.5

$$
\begin{aligned}
f_{\bar{\alpha}_j}(s_j, r_j; [\boldsymbol{\theta}_I]_j) = \pi_j^{\mathrm{w}}(r_j, s_j, [\boldsymbol{\theta}_I]_j)\Bigg\{ \\
\mu_j + \frac{\underline{\mathcal{Z}}_{I_j}}{\mathcal{Z}_{\varphi_j}} \left(\underline{r}_j - \sqrt{s_j} \frac{\phi_{\mathcal{N}}\left(\frac{-\underline{r}_j}{\sqrt{s_j}}\right)}{\Phi_{\mathcal{N}}\left(\frac{-\underline{r}_j}{\sqrt{s_j}}\right)} \right) + \frac{\bar{\mathcal{Z}}_{I_j}}{\mathcal{Z}_{\varphi_j}} \left(\bar{r}_j + \sqrt{s_j} \frac{\phi_{\mathcal{N}}\left(\frac{\bar{r}_j}{\sqrt{s_j}}\right)}{\Phi_{\mathcal{N}}\left(\frac{\bar{r}_j}{\sqrt{s_j}}\right)} \right) \Bigg\}
\end{aligned}
\tag{3.145}
$$

$$
\begin{aligned}
f_{\tilde{\alpha}_j}(s_j, r_j; [\boldsymbol{\theta}_I]_j) = 2\mu f_{\bar{\alpha}_j}(s_j, r_j; [\boldsymbol{\theta}_I]_j) + \pi_j^{\mathrm{w}}(r_j, s_j, [\boldsymbol{\theta}_I]_j)\Bigg\{ - \mu_j^2 \\
+ \frac{\underline{\mathcal{Z}}_{I_j}}{\mathcal{Z}_{\varphi_j}} \left[s_j \left(1 - \frac{\phi_{\mathcal{N}}\left(\frac{-\underline{r}_j}{\sqrt{s_j}}\right)}{\Phi_{\mathcal{N}}\left(\frac{-\underline{r}_j}{\sqrt{s_j}}\right)} \left(\frac{\phi_{\mathcal{N}}\left(\frac{-\underline{r}_j}{\sqrt{s_j}}\right)}{\Phi_{\mathcal{N}}\left(\frac{-\underline{r}_j}{\sqrt{s_j}}\right)} - \frac{\underline{r}_j}{\sqrt{s_j}} \right) \right) + \left(\underline{r}_j - \sqrt{s_j} \frac{\phi_{\mathcal{N}}\left(\frac{-\underline{r}_j}{\sqrt{s_j}}\right)}{\Phi_{\mathcal{N}}\left(\frac{-\underline{r}_j}{\sqrt{s_j}}\right)} \right)^2 \right] \\
+ \frac{\bar{\mathcal{Z}}_{I_j}}{\mathcal{Z}_{\varphi_j}} \left[s_j \left(1 - \frac{\phi_{\mathcal{N}}\left(\frac{\bar{r}_j}{\sqrt{s_j}}\right)}{\Phi_{\mathcal{N}}\left(\frac{\bar{r}_j}{\sqrt{s_j}}\right)} \left(\frac{\phi_{\mathcal{N}}\left(\frac{\bar{r}_j}{\sqrt{s_j}}\right)}{\Phi_{\mathcal{N}}\left(\frac{\bar{r}_j}{\sqrt{s_j}}\right)} + \frac{\bar{r}_j}{\sqrt{s_j}} \right) \right) + \left(\bar{r}_j + \sqrt{s_j} \frac{\phi_{\mathcal{N}}\left(\frac{\bar{r}_j}{\sqrt{s_j}}\right)}{\Phi_{\mathcal{N}}\left(\frac{\bar{r}_j}{\sqrt{s_j}}\right)} \right)^2 \right] \Bigg\} \\
- f_{\bar{\alpha}_j}(s_j, r_j; [\boldsymbol{\theta}_I]_j)^2
\end{aligned}
\tag{3.146}
$$

for

$$\mathcal{Z}_{\varphi_j} = \underline{\mathcal{Z}}_{I_j} + \bar{\mathcal{Z}}_{I_j} \tag{3.147}$$

$$\underline{\mathcal{Z}}_{I_j} = \frac{\lambda_j}{2} \exp\left(\frac{1}{2}\lambda_j^2 s_j + \check{r}_j \lambda_j \right) \Phi_{\mathcal{N}}\left(\frac{-\underline{r}_j}{\sqrt{s_j}} \right) \tag{3.148}$$

$$\bar{\mathcal{Z}}_{I_j} = \frac{\lambda_j}{2} \exp\left(\frac{1}{2}\lambda_j^2 s_j - \check{r}_j \lambda_j \right) \Phi_{\mathcal{N}}\left(\frac{\bar{r}_j}{\sqrt{s_j}} \right) \tag{3.149}$$

$$\check{r}_j = r_j - \mu_j \tag{3.150}$$

$$\underline{r}_j = \check{r}_j + \lambda_j s_j \tag{3.151}$$

$$\bar{r}_j = \check{r}_j - \lambda_j s_j \tag{3.152}$$

3.7.2 I.i.d. Sparse Bernoulli-Gauss Input Channel

For an i.i.d. BG input channel with signal density τ and Gaussian mean $\bar{\theta}$ and variance $\tilde{\theta}$ ($\boldsymbol{\theta}_I = [\tau, \bar{\theta}, \tilde{\theta}]^T$), i.e.

$$p(\alpha; \boldsymbol{\theta}_I) = (1 - \tau)\delta_{\mathrm{Dirac}}(\alpha) + \tau \frac{1}{\sqrt{2\pi\tilde{\theta}}} \exp\left(-\frac{(\alpha - \bar{\theta})^2}{2\tilde{\theta}} \right) \tag{3.153}$$

we have channel functions [5][3] (Eqs. (68), (69))

$$f_{\bar{\alpha}}(s, r; \boldsymbol{\theta}_I) = \frac{\tau a b \frac{1}{s + \tilde{\theta}} c}{(1 - \tau) d + \tau b a} \tag{3.154}$$

$$f_{\tilde{\alpha}}(s, r; \boldsymbol{\theta}_I) = \frac{\tau(1 - \tau) d a b \frac{1}{(s + \tilde{\theta})^2} (\tilde{\theta} s (s + \tilde{\theta}) + c^2) + \tau^2 a \tilde{\theta} b^4}{((1 - \tau) d + \tau b a)^2} \tag{3.155}$$

for

$$a := \exp\left(-\frac{(r - \bar{\theta})^2}{2(s + \tilde{\theta})} \right) \tag{3.156}$$

[3] In [5] the full expressions for the channel functions are given, i.e. the intermediate variables a, b, c, d are not used. These variables have been introduced by the authors of this note.

$$\mathscr{b} := \frac{\sqrt{s}}{\sqrt{s+\tilde{\theta}}} \tag{3.157}$$

$$\mathscr{c} := \bar{\theta}s + r\tilde{\theta} \tag{3.158}$$

$$\mathscr{d} := \exp\left(-\frac{r^2}{2s}\right) \tag{3.159}$$

Manoel and Tramel suggested the following implementation[4] of the i.i.d. BG input channel functions

$$f_{\bar{\alpha}}(s,r;\boldsymbol{\theta}_I) = \frac{\frac{\mathscr{c}}{s+\tilde{\theta}}}{\mathscr{f}+1} \tag{3.160}$$

$$f_{\tilde{\alpha}}(s,r;\boldsymbol{\theta}_I) = \mathscr{f} f_{\bar{\alpha}}^2(s,r;\boldsymbol{\theta}_I) + \frac{\mathscr{e}}{\mathscr{f}+1} \tag{3.161}$$

for

$$\mathscr{e} := \frac{\tilde{\theta}s}{s+\tilde{\theta}} \left[= \tilde{\theta}\mathscr{b}^2\right] \tag{3.162}$$

$$\mathscr{f} := \frac{1-\tau}{\tau}\sqrt{\frac{\tilde{\theta}}{\mathscr{e}}}\exp\left(-\frac{1}{2}\left(\frac{r^2}{s} - \frac{(r-\bar{\theta})^2}{s+\tilde{\theta}}\right)\right) \left[= \frac{1-\tau}{\tau}\frac{1}{\mathscr{b}}\frac{\mathscr{d}}{a}\right] \tag{3.163}$$

Parker suggested the following implementation of the i.i.d. BG input channel functions [15]

$$f_{\bar{\alpha}}(s,r;\boldsymbol{\theta}_I) = \frac{\mathscr{g}}{\mathscr{k}} \tag{3.164}$$

$$f_{\tilde{\alpha}}(s,r;\boldsymbol{\theta}_I) = \mathscr{g}^2\frac{\mathscr{k}-1}{\mathscr{k}^2} + \frac{\mathscr{h}}{\mathscr{k}} \tag{3.165}$$

for

$$\mathscr{g} := \frac{\frac{\bar{\theta}}{\tilde{\theta}} + \frac{r}{s}}{\frac{1}{\tilde{\theta}} + \frac{1}{s}} \left[= \frac{\mathscr{c}}{s+\tilde{\theta}}\right] \tag{3.166}$$

$$\mathscr{h} := \frac{1}{\frac{1}{\tilde{\theta}} + \frac{1}{s}} \left[= \mathscr{e}\right] \tag{3.167}$$

$$\mathscr{k} := 1 + \frac{1-\tau}{\tau}\sqrt{\frac{\tilde{\theta}}{\mathscr{h}}}\exp\left(\frac{1}{2}\left[\frac{(r-\bar{\theta})^2}{\tilde{\theta}+s} - \frac{r^2}{s}\right]\right) \left[= 1 + \frac{1-\tau}{\tau}\frac{1}{\mathscr{b}}\frac{\mathscr{d}}{a}\right] \tag{3.168}$$

However, in EM-BG-GAMP [39], the following slightly modified implementation of the i.i.d. BG input channel is suggested

$$f_{\bar{\alpha}}(s,r;\boldsymbol{\theta}_I) = \mathscr{l}\mathscr{g} \left[= \frac{\mathscr{g}}{\mathscr{k}}\right] \tag{3.169}$$

$$f_{\tilde{\alpha}}(s,r;\boldsymbol{\theta}_I) = \mathscr{l}(\mathscr{h} + |\mathscr{g}|^2) - \mathscr{l}^2|\mathscr{g}|^2 \left[= \frac{\mathscr{h} + |\mathscr{g}|^2}{\mathscr{k}} - \frac{|\mathscr{g}|^2}{\mathscr{k}^2} = \frac{\mathscr{k}\mathscr{h} + |\mathscr{g}|^2(\mathscr{k}-1)}{\mathscr{k}^2} = |\mathscr{g}|^2\frac{\mathscr{k}-1}{\mathscr{k}^2} + \frac{\mathscr{h}}{\mathscr{k}}\right] \tag{3.170}$$

for

$$\mathscr{l} := \frac{1}{1 + \left(\frac{\tau}{1-\tau}\frac{\frac{1}{\sqrt{\tilde{\theta}+s}}\exp\left(-\frac{1}{2}\frac{(r-\bar{\theta})^2}{\tilde{\theta}+s}\right)}{\frac{1}{\sqrt{s}}\exp\left(-\frac{1}{2}\frac{r^2}{s}\right)}\right)^{-1}} \left[= \frac{1}{1 + \left(\frac{\tau}{1-\tau}\mathscr{b}\frac{a}{d}\right)^{-1}} = \frac{1}{\mathscr{k}}\right] \tag{3.171}$$

Note that the $\mathscr{l}$ in Equation (3.171) corresponds to the $\pi(r,s,\boldsymbol{\theta}_I)$ in Equation (3.41). Also note that the absolute values are included for generality since the case of real valued Gaussians may be extended to the case of circular-complex-Gaussians [39], [28].

[4] See: `https://github.com/eric-tramel/SwAMP-Demo/blob/master/python/amp.py`

3.7.2.1 BG Expressions for use with the GWS Input Channel

Based on the result in Equations (3.170) and (3.170) as well as the multiplication rule for two Gaussian densities given in [28], we may identify the following i.i.d. BG channel updates to be used in the GWS framework described in Section 3.5

$$f_{\bar{\alpha}_j}(s_j, r_j; [\boldsymbol{\theta}_I]_j) = \pi_j^{\mathrm{w}}(r_j, s_j, [\boldsymbol{\theta}_I]_j) \left(\frac{\frac{\bar{\theta}_j}{\tilde{\theta}_j} + \frac{r_j}{s_j}}{\frac{1}{\tilde{\theta}_j} + \frac{1}{s_j}} \right) \tag{3.172}$$

$$f_{\tilde{\alpha}_j}(s_j, r_j; [\boldsymbol{\theta}_I]_j) = \pi_j^{\mathrm{w}}(r_j, s_j, [\boldsymbol{\theta}_I]_j) \left(\frac{1}{\frac{1}{\tilde{\theta}_j} + \frac{1}{s_j}} + \left(\frac{\frac{\bar{\theta}_j}{\tilde{\theta}_j} + \frac{r_j}{s_j}}{\frac{1}{\tilde{\theta}_j} + \frac{1}{s_j}} \right)^2 \right) - f_{\bar{\alpha}_j}(s_j, r_j; [\boldsymbol{\theta}_I]_j)^2 \tag{3.173}$$

4 Sum Approximations

From an implementation point of view, a critical element in the GAMP iteration in Equations (2.1)-(2.12) is the application of the entrywise absolute value squared system matrix

$$|\mathbf{A}|^{\circ 2} = \mathbf{A}_{\text{asq}} = \begin{bmatrix} |a_{11}|^2 & \cdots & |a_{1n}|^2 \\ \vdots & \ddots & \vdots \\ |a_{m1}|^2 & \cdots & |a_{mn}|^2 \end{bmatrix} \tag{4.1}$$

If the system matrix $\mathbf{A}$ is given explicitly, one may easily find $|\mathbf{A}|^{\circ 2}$ using Equation (4.1). However, oftentimes (especially when considering large problem sizes) it is a necessity to use a *fast transform* for implementing the matrix-vector products involving $\mathbf{A}$ and $\mathbf{A}^H$ in the GAMP algorithm in order to achieve acceptable reconstruction times and reasonable memory requirements [28], [29]. For instance, one may use a Fast Fourier Transform (FFT) based method to implement a matrix-vector product involving a Discrete Fourier Transform (DFT). However, such fast transforms are not always available for implementing the matrix-vector products involving $|\mathbf{A}|^{\circ 2}$. As an alternative, one may use sum approximation (also known as uniform variance) GAMP. The idea, then, is to approximate the matrix-vector products involving $|\mathbf{A}|^{\circ 2}$ by certain sums.

4.1 The Sum Approximation by Krzakala et al.

In [5], Krzakala et al. consider the case where $\mathbf{A}$ is a homogeneous matrix with i.i.d. random entries having zero mean and variance $\frac{1}{n}$. In this case, the ensemble average of different realisations of $|\mathbf{A}|^{\circ 2}$ is

$$\begin{aligned}
\mathbb{E}[|\mathbf{A}|^{\circ 2}] &= \begin{bmatrix} \mathbb{E}[|a_{11}|^2] & \cdots & \mathbb{E}[|a_{1n}|^2] \\ \vdots & \ddots & \vdots \\ \mathbb{E}[|a_{m1}|^2] & \cdots & \mathbb{E}[|a_{mn}|^2] \end{bmatrix} && (4.2)\\
&= \begin{bmatrix} \mathrm{Var}(a_{11}) & \cdots & \mathrm{Var}(a_{1n}) \\ \vdots & \ddots & \vdots \\ \mathrm{Var}(a_{m1}) & \cdots & \mathrm{Var}(a_{mn}) \end{bmatrix} && (4.3)\\
&= \begin{bmatrix} \frac{1}{n} & \cdots & \frac{1}{n} \\ \vdots & \ddots & \vdots \\ \frac{1}{n} & \cdots & \frac{1}{n} \end{bmatrix} && (4.4)
\end{aligned}$$

since the entries of $\mathbf{A}$ are zero mean. Now, one may consider an approximation of e.g. the GAMP factor side update in Equation (2.1)

$$\begin{aligned}
\bar{v}^{t+1} &= \sum_j \mathbb{E}[|\mathbf{A}|^{\circ 2}]_{ij} \tilde{\alpha}_j^t && (4.5)\\
&= \frac{1}{n} \sum_j \tilde{\alpha}_j^t && (4.6)
\end{aligned}$$

Since the variance $\overline{(v_i^{t+1} - \bar{v}^{t+1})^2}$ is of order $\mathcal{O}\left(\frac{1}{n}\right)$ (see the details in Eq. (55) in [5]), in the large system limit as $n \to \infty$, one may consider all v_i^{t+1} equal to their average $\bar{v}^{t+1}$. Similar arguments may be used for other GAMP updates that involve $|\mathbf{A}|^{\circ 2}$. Thus, one may replace all GAMP instances of $|\mathbf{A}|^{\circ 2}$ with sums scaled by $\frac{1}{n}$, which yields the following alternatives to the GAMP updates in Equations (2.1) and (2.7)

$$\bar{v}^{t+1} = \frac{1}{n}\sum_j \tilde{\alpha}_j^t \tag{4.7}$$

$$\bar{s}^{t+1} = \left[\frac{1}{n}\sum_i u_i^{t+1}\right]^{-1} \tag{4.8}$$

which may then be used in place of all v_i^{t+1}, s_j^{t+1} respectively, in the GAMP iteration in Equations (2.1)-(2.12). Note that the sum approximation by Krzakala et al. (with an assumed variance of $\frac{1}{m}$ instead of $\frac{1}{n}$) is closely related to the Donoho/Maleki/Montanari AMP as described in Section 2.1.

4.2 The Sum Approximation by Rangan

In [16], Rangan considers a sightly different approximation based on the assumption that $|a_{ij}|^2 \approx \frac{||\mathbf{A}||_F^2}{mn}$ for all i, j, where $||\mathbf{A}||_F^2$ is the Frobenius norm of the matrix $\mathbf{A}$. Rangan then forces all variance related components for each GAMP state to be the same. That is, $v_i^{t+1} = \breve{v}^{t+1}$, $u_i^{t+1} = \breve{u}^{t+1}$, $s_j^{t+1} = \breve{s}^{t+1}$, $\tilde{\alpha}_j^{t+1} = \breve{\alpha}^{t+1}$ for all i, j. Specifically, Rangan's MMSE GAMP iteration with scalar variances reads

Output (factor) side updates:

$$\breve{v}^{t+1} = \frac{1}{m}||\mathbf{A}||_F^2 \breve{\alpha}^t \tag{4.9}$$

$$o_i^{t+1} = \sum_j a_{ij}\bar{\alpha}_j^t - \breve{v}^{t+1} q_i^t \tag{4.10}$$

$$\bar{z}_i^{t+1} = f_{\bar{z}_i}(\breve{v}^{t+1}, o_i^{t+1}; y_i, [\boldsymbol{\theta}_o]_i^t) \tag{4.11}$$

$$\tilde{z}_i^{t+1} = f_{\tilde{z}_i}(\breve{v}^{t+1}, o_i^{t+1}; y_i, [\boldsymbol{\theta}_o]_i^t) \tag{4.12}$$

$$q_i^{t+1} = \frac{\bar{z}_i^{t+1} - o_i^{t+1}}{\breve{v}^{t+1}} \tag{4.13}$$

$$\breve{u}^{t+1} = \frac{1}{m}\sum_i \frac{\breve{v}^{t+1} - \tilde{z}_i^{t+1}}{(\breve{v}^{t+1})^2} \tag{4.14}$$

Input (variable) side updates:

$$\breve{s}^{t+1} = \left[\frac{1}{n}||\mathbf{A}||_F^2 \breve{u}^{t+1}\right]^{-1} \tag{4.15}$$

$$r_j^{t+1} = \bar{\alpha}_j^t + \breve{s}^{t+1}\sum_i a_{ij}^* q_i^{t+1} \tag{4.16}$$

$$\bar{\alpha}_j^{t+1} = f_{\bar{\alpha}_j}(\breve{s}^{t+1}, r_j^{t+1}; [\boldsymbol{\theta}_I]_j^t) \tag{4.17}$$

$$\breve{\alpha}^{t+1} = \frac{1}{n}\sum_j f_{\tilde{\alpha}_j}(\breve{s}^{t+1}, r_j^{t+1}; [\boldsymbol{\theta}_I]_j^t) \tag{4.18}$$

Optional parameter value updates (using e.g. EM - see also Section 6):

$$[\boldsymbol{\theta}_o]_i^{t+1} = \dots \tag{4.19}$$

$$[\boldsymbol{\theta}_I]_j^{t+1} = \dots \tag{4.20}$$

In [16], Rangan claims that simulations show that this simplified GAMP iteration works as well as the full (non-uniform variance) GAMP iteration in Equations (2.1) - (2.12). Unfortunately, neither the specific simulations nor any details about their nature are given in [16].

4.2.1 The Connection Between Sum Approximations by Krzakala et al. and Rangan

Rangan's simplifications are closely related to simplifications by Krzakala et al. To see this, note that for an $m \times n$ system matrix $\mathbf{A}$ with i.i.d. random entries having zero mean and variance σ^2, we have

$$\sigma^2 \approx \frac{1}{mn} \sum_i \sum_j |a_{ij}|^2 = \frac{1}{mn} ||\mathbf{A}||_F^2 \tag{4.21}$$

Thus, if $\sigma^2 = \frac{1}{n}$ as in the sum approximation by Krzakala et al. and $v_i^{t+1} \approx \breve{v}^{t+1}$, $s_j^{t+1} \approx \breve{s}^{t+1}$ for all $i = 1, \ldots, m$ and $j = 1, \ldots, n$, we have

$$\begin{aligned}
\bar{v}^{t+1} &= \sigma^2 \sum_j \tilde{\alpha}_j^t && (4.22)\\
&\approx \frac{1}{mn} ||\mathbf{A}||_F^2 \sum_j \tilde{\alpha}_j^t && (4.23)\\
&\approx \frac{1}{m} ||\mathbf{A}||_F^2 \breve{\alpha}^t && (4.24)\\
&= \breve{v}^{t+1} && (4.25)\\
\bar{s}^{t+1} &= \left[\sigma^2 \sum_i u_i^{t+1} \right]^{-1} && (4.26)\\
&\approx \left[\frac{1}{mn} ||\mathbf{A}||_F^2 \sum_i u_i^{t+1} \right]^{-1} && (4.27)\\
&\approx \left[\frac{1}{n} ||\mathbf{A}||_F^2 \breve{u}^{t+1} \right]^{-1} && (4.28)\\
&= \breve{s}^{t+1} && (4.29)
\end{aligned}$$

An interpretation of this result is that Rangan's sum approximation adapts to the system matrix $\mathbf{A}$ (and its variance) through $||\mathbf{A}||_F^2$.

4.2.2 Efficiently Computing the Frobenius Norm of the System Matrix

The need for finding $||\mathbf{A}||_F^2$ may make it infeasible to use Rangan's sum approximation in practical applications. For instance, if one attempts to use Rangan's method in order to avoid explicitly storing $|\mathbf{A}|^{\circ 2}$ in memory on a computer because it is infeasible to do so, a method for implicitly finding $||\mathbf{A}||_F^2$ is needed. If no such method is available and one has to explicitly store $\mathbf{A}$ in memory in order to estimate $||\mathbf{A}||_F^2$, no progress is made in using Rangan's sum approximation. We now discuss a few structured system matrices that allow for efficient (in terms of memory and computation requirements) ways to compute $||\mathbf{A}||_F^2$.

If $\mathbf{A}$ is defined by a Kronecker product, i.e. $\mathbf{A} = \mathbf{B} \otimes \mathbf{C} \in \mathbb{C}^{m \times n}$ for $\mathbf{B} \in \mathbb{C}^{o \times p}$, $\mathbf{C} \in \mathbb{C}^{q \times r}$ with $m = o \cdot q$, $n = p \cdot r$, then we have

$$\begin{aligned}
||\mathbf{A}||_F^2 &= \sum_{i'=0}^{m-1} \sum_{j'=0}^{n-1} |a_{i'j'}|^2 && (4.30)\\
&= \sum_{i=0}^{o-1} \sum_{j=0}^{p-1} \sum_{k=0}^{q-1} \sum_{l=0}^{r-1} |b_{ij} c_{kl}|^2 && (4.31)\\
&= \sum_{i=0}^{o-1} \sum_{j=0}^{p-1} \sum_{k=0}^{q-1} \sum_{l=0}^{r-1} |b|_{ij}^2 |c|_{kl}^2 && (4.32)\\
&= \left(\sum_{i=0}^{o-1} \sum_{j=0}^{p-1} |b|_{ij}^2 \right) \left(\sum_{k=0}^{q-1} \sum_{l=0}^{r-1} |c|_{kl}^2 \right) && (4.33)\\
&= ||\mathbf{B}||_F^2 ||\mathbf{C}||_F^2 && (4.34)
\end{aligned}$$

Using an inductive argument, it is easy to see from the above computations that this property of the Frobenius norm generalises to Kronecker products of more than two matrices.

Now consider all matrices that are arbitrary-sign adjusted, permuted, and scaled identity matrices, i.e. any matrix $\mathbf{G} \in \mathbb{C}^{n \times n}$ formed from an identity matrix by scaling all diagonal entries by an arbitrary factor $b \in \mathbb{C}$, followed by arbitrary sign changes on the diagonal entries, followed by an arbitrary permutation of either rows and/or columns. Now since we are only moving around and scaling (with a common absolute value of the scaling factor) all entries of the identity matrix, we have the following property of the Frobenius norm of the matrix products $\mathbf{H}_1\mathbf{G}$, $\mathbf{G}\mathbf{H}_2$ for any $\mathbf{H}_1 \in \mathbb{C}^{m \times n}$, $\mathbf{H}_2 \in \mathbb{C}^{n \times m}$:

$$
\begin{aligned}
||\mathbf{H}_1\mathbf{G}||_F^2 &= \sum_{i'=0}^{m-1}\sum_{j'=0}^{n-1} |h_{1_{i'j'}} \cdot b|^2 && (4.35)\\
&= |b|^2 \sum_{i'=0}^{m-1}\sum_{j'=0}^{n-1} |h_{1_{i'j'}}|^2 && (4.36)\\
&= |b|^2 \cdot ||\mathbf{H}_1||_F^2 && (4.37)\\
||\mathbf{G}\mathbf{H}_2||_F^2 &= \sum_{i''=0}^{n-1}\sum_{j''=0}^{m-1} |b \cdot h_{1_{i''j''}}|^2 && (4.38)\\
&= |b|^2 \cdot \sum_{i''=0}^{n-1}\sum_{j''=0}^{m-1} |h_{1_{i''j''}}|^2 && (4.39)\\
&= |b|^2 \cdot ||\mathbf{H}_2||_F^2 && (4.40)\\
& && (4.41)
\end{aligned}
$$

with the i', j' indexing $\mathbf{H}_1$ according to the permutations applied by $\mathbf{G}$ and the i'', j'' indexing $\mathbf{H}_2$ according to the permutations applied by $\mathbf{G}$.

Finally, we consider the Structurally Random Matrices (SRMs) detailed in [30]. These matrices are defined by

$$
\mathbf{A} = \mathbf{D}_\Omega \mathbf{F} \mathbf{R} \qquad (4.42)
$$

with $\mathbf{D}_\Omega \in \mathbb{R}^{m \times n}$ a sub-sampling matrix that selects a (random) subset of the rows from $\mathbf{FR}$ according to the indexing set Ω, i.e. it is an identity matrix with the rows not indexed by Ω removed, $\mathbf{F} \in \mathbb{R}^{n \times n}$ an orthogonal matrix, and $\mathbf{R} \in \mathbb{R}^{n \times n}$ a prerandomization matrix, i.e. either an identity matrix with uniformly random sign changes on the diagonal entries or a permutation matrix that permutes the columns of $\mathbf{F}$ at (uniformly) random. From Equation (4.37), we have $||\mathbf{FR}||_F^2 = ||\mathbf{F}||_F^2$. The sub-sampling by $\mathbf{D}_\Omega$ results in keeping only a fraction of $\frac{m}{n}$ of the (unit vector) rows in $\mathbf{FR}$. Thus, for $\mathbf{A}$ a SRM, we get

$$
\begin{aligned}
||\mathbf{A}||_F^2 &= ||\mathbf{D}_\Omega \mathbf{F} \mathbf{R}||_F^2 && (4.43)\\
&= \frac{m}{n}||\mathbf{FR}||_F^2 && (4.44)\\
&= \frac{m}{n}||\mathbf{F}||_F^2 && (4.45)\\
&= \frac{m}{n}n && (4.46)\\
&= m && (4.47)
\end{aligned}
$$

If $\mathbf{F}$ is not an orthogonal matrix, Equation (4.45) is not in general valid. However, if we assume that all entries of $|\mathbf{F}|^{\circ 2}$ are of approximately the same size, i.e. $|f_{ij}|^2 \approx \frac{||\mathbf{F}||_F^2}{n^2}$ for all i, j (essentially the same assumption that is used in Rangan's sum approximation), we have

$$
||\mathbf{A}||_F^2 \approx \frac{m}{n}||\mathbf{F}||_F^2 \qquad (4.48)
$$

for any such $\mathbf{F}$. Furthermore, if $\mathbf{F}$ is defined by a Kronecker product, we may use Equation (4.34) to compute $||\mathbf{F}||_F^2$.

5 Implementations of the GAMP Iteration

The GAMP iteration given in Equations (2.1)-(2.12) may be implemented in a number of ways. One may elect to combine some of the states, introduce new ones, or introduce convergence supporting heuristics. In this section, we present some of the implementations of the GAMP iteration that may be found in the literature.

The algorithms presented in this section are described using matrix-vector notation with Numpy broadcasting rules[1], e.g. multiplying two (column) vectors amounts to entrywise multiplication. The matrix-vector notation takes precedence over broadcasting[2], e.g. multiplying a matrix with a vector amounts to a matrix-vector multiplication - not a broadcast along the last axis. All vectors are column vectors. Also $\boldsymbol{\theta}_I$ and $\boldsymbol{\theta}_o$ are to be understood as being the relevant channel parameters for the given iteration as detailed in Sections 3.1 and 3.2. The channel functions $f_{\bar{z}}$, $f_{\tilde{z}}$, $f_{\bar{\alpha}}$, and $f_{\tilde{\alpha}}$ are scalar functions as noted in Section 3. Thus, when used on vectors these functions operate on each element of the vectors and produce a result vector having the individual scalar results as its entries. Consequently, the result is a vector of the same size as the input vectors.

Algorithm 1 details the implementation of the MMSE GAMP from [1], [16] as described by Parker [15]. Algorithm 1 is the GAMP variant used in Schniter's and Vila's EM-BG-AMP and EM-GM-AMP algorithms [39], [45], [28].

Algorithm 1 - MMSE GAMP [1], [15]

```
initialise: $\bar{\boldsymbol{\alpha}}_0 = \mathbb{E}_{\boldsymbol{\alpha}|\boldsymbol{\theta}_I}[\boldsymbol{\alpha}]$, $\tilde{\boldsymbol{\alpha}}_0 = \mathrm{Var}_{\boldsymbol{\alpha}|\boldsymbol{\theta}_I}(\boldsymbol{\alpha})$, $\mathbf{q}_0 = \mathbf{0}_m$    # marginal conditional expectations
for $t = 1 \dots T_{\max}$ do
    $\mathbf{v}_t = \mathbf{A}_{\mathrm{asq}}\tilde{\boldsymbol{\alpha}}_{t\text{-}1}$
    $\mathbf{o}_t = \mathbf{A}\bar{\boldsymbol{\alpha}}_{t\text{-}1} - \mathbf{v}_t\mathbf{q}_{t\text{-}1}$
    $\bar{\mathbf{z}}_t = f_{\bar{z}}(\mathbf{v}_t, \mathbf{o}_t; \mathbf{y}, \boldsymbol{\theta}_o)$
    $\tilde{\mathbf{z}}_t = f_{\tilde{z}}(\mathbf{v}_t, \mathbf{o}_t; \mathbf{y}, \boldsymbol{\theta}_o)$
    $\mathbf{q}_t = \frac{\bar{\mathbf{z}}_t - \mathbf{o}_t}{\mathbf{v}_t}$
    $\mathbf{u}_t = \frac{\mathbf{v}_t - \tilde{\mathbf{z}}_t}{\mathbf{v}_t^2}$
    $\mathbf{s}_t = [\mathbf{A}_{\mathrm{asq}}^T\mathbf{u}_t]^{-1}$
    $\mathbf{r}_t = \bar{\boldsymbol{\alpha}}_{t\text{-}1} + \mathbf{s}_t\mathbf{A}^H\mathbf{q}_t$
    $\bar{\boldsymbol{\alpha}}_t = f_{\bar{\alpha}}(\mathbf{s}_t, \mathbf{r}_t; \boldsymbol{\theta}_I)$
    $\tilde{\boldsymbol{\alpha}}_t = f_{\tilde{\alpha}}(\mathbf{s}_t, \mathbf{r}_t; \boldsymbol{\theta}_I)$
    if stop criterion is met then
        break
    end if
end for
```

Parker introduced some further modifications for numerical robustness of the GAMP algorithm [15]. This modified algorithm is detailed in Algorithm 2. Compared to Algorithm 1, two modifications are made:

- Introduction of a step-size (or damping) parameter κ.
- Re-scaling of several of the states to better handle high SNR cases.

The simplified sum approximation GAMP algorithms described Section 4 have similar implementations to the algorithms presented so far. In particular, the sum approximation by Krzakala et al. in Equations (4.7) and (4.8) may be implemented in Algorithm 1 be replacing $\mathbf{A}_{\mathrm{asq}}$ with $\frac{1}{n}\mathbf{1}_n^T$ and replacing $\mathbf{A}_{\mathrm{asq}}^T$ with $\frac{1}{n}\mathbf{1}_m^T$ which means that $\mathbf{v}_t$ and $\mathbf{s}_t$ become scalar. Rangan's simplified sum

[1] See: `http://docs.scipy.org/doc/numpy-1.10.1/user/basics.broadcasting.html`

[2] The way to think of it: Whenever you encounter an undefined operation in matrix-vector notation, then use the broadcasting rules.

Algorithm 2 - Numerically Robust MMSE GAMP with damping [15]

```
**initialise:** $\bar{\boldsymbol{\alpha}}_0 = \mathbb{E}_{\boldsymbol{\alpha}|\boldsymbol{\theta}_I}[\boldsymbol{\alpha}]$, $\tilde{\boldsymbol{\alpha}}_0 = \mathrm{Var}_{\boldsymbol{\alpha}|\boldsymbol{\theta}_I}(\boldsymbol{\alpha})$, $\mathbf{q}_0 = \mathbf{0}_m$, $\mu_0 = 1$, $\mathbf{v}_0 = \mathbf{0}_m$    # marginal conditional expectations
**for** $t = 1 \dots T_{\max}$ **do**
    $\mathbf{v}_t = \kappa \mathbf{A}_{\text{asq}} \tilde{\boldsymbol{\alpha}}_{t\text{-}1} + (1-\kappa)\mathbf{v}_{t\text{-}1}$
    $\mu_t = \frac{1}{m}\mathbf{v}_t^{\mathrm{T}} \mathbf{1}_m$
    $\mathbf{o}_t = \mathbf{A}\bar{\boldsymbol{\alpha}}_{t\text{-}1} - \frac{1}{\mu_t}\mathbf{v}_t \mathbf{q}_{t\text{-}1}$
    $\bar{\mathbf{z}}_t = f_{\bar{z}}(\mathbf{v}_t, \mathbf{o}_t; \mathbf{y}, \boldsymbol{\theta}_o)$
    $\tilde{\mathbf{z}}_t = f_{\tilde{z}}(\mathbf{v}_t, \mathbf{o}_t; \mathbf{y}, \boldsymbol{\theta}_o)$
    $\mathbf{q}_t = \kappa \mu_t \frac{\bar{\mathbf{z}}_t - \mathbf{o}_t}{\mathbf{v}_t} + (1-\kappa)\mathbf{q}_{t\text{-}1}$
    $\mathbf{u}_t = \kappa \mu_t \frac{\mathbf{v}_t - \tilde{\mathbf{z}}_t}{\mathbf{v}_t^2} + (1-\kappa)\mathbf{u}_{t\text{-}1}$
    $\breve{\boldsymbol{\alpha}}_t = \kappa \bar{\boldsymbol{\alpha}}_{t\text{-}1} + (1-\kappa)\breve{\boldsymbol{\alpha}}_{t\text{-}1}$
    $\mathbf{s}_t = [\mathbf{A}_{\text{asq}}^T \mathbf{u}_t]^{-1}$
    $\mathbf{r}_t = \breve{\boldsymbol{\alpha}}_t + \mathbf{s}_t \mathbf{A}^H \mathbf{q}_t$
    $\bar{\boldsymbol{\alpha}}_t = f_{\bar{\alpha}}(\mathbf{s}_t, \mathbf{r}_t; \boldsymbol{\theta}_I)$
    $\tilde{\boldsymbol{\alpha}}_t = \mu_t f_{\tilde{\alpha}}(\mathbf{s}_t, \mathbf{r}_t; \boldsymbol{\theta}_I)$
    **if** stop criterion is met **then**
        **break**
    **end if**
**end for**
```

approximation GAMP iteration in Equations (4.9)-(4.20) may be implemented in a similar way to Algorithm 1 as detailed in Algorithm 3.

Algorithm 3 - MMSE GAMP with Rangan sum approximations [16]

```
**initialise:** $\bar{\boldsymbol{\alpha}}_0 = \mathbb{E}_{\boldsymbol{\alpha}|\boldsymbol{\theta}_I}[\boldsymbol{\alpha}]$, $\breve{\alpha}_0 = \frac{1}{n}\sum\left(\mathrm{Var}_{\boldsymbol{\alpha}|\boldsymbol{\theta}_I}(\boldsymbol{\alpha})\right)$, $\mathbf{q}_0 = \mathbf{0}_m$    # marginal conditional expectations
**for** $t = 1 \dots T_{\max}$ **do**
    $\breve{v}_t = \frac{1}{m}||\mathbf{A}||_F^2 \breve{\alpha}_{t\text{-}1}$
    $\mathbf{o}_t = \mathbf{A}\bar{\boldsymbol{\alpha}}_{t\text{-}1} - \breve{v}_t \mathbf{q}_{t\text{-}1}$
    $\bar{\mathbf{z}}_t = f_{\bar{z}}(\breve{v}_t, \mathbf{o}_t; \mathbf{y}, \boldsymbol{\theta}_o)$
    $\tilde{\mathbf{z}}_t = f_{\tilde{z}}(\breve{v}_t, \mathbf{o}_t; \mathbf{y}, \boldsymbol{\theta}_o)$
    $\mathbf{q}_t = \frac{\bar{\mathbf{z}}_t - \mathbf{o}_t}{\breve{v}_t}$
    $\breve{u}_t = \frac{1}{m}\sum\left(\frac{\breve{v}_t - \tilde{\mathbf{z}}_t}{\breve{v}_t^2}\right)$
    $\breve{s}_t = [\frac{1}{n}||\mathbf{A}||_F^2 \breve{u}_t]^{-1}$
    $\mathbf{r}_t = \bar{\boldsymbol{\alpha}}_{t\text{-}1} + \breve{s}_t \mathbf{A}^H \mathbf{q}_t$
    $\bar{\boldsymbol{\alpha}}_t = f_{\bar{\alpha}}(\breve{s}_t, \mathbf{r}_t; \boldsymbol{\theta}_I)$
    $\breve{\alpha}_t = \frac{1}{n}\sum\left(f_{\tilde{\alpha}}(\breve{s}_t, \mathbf{r}_t; \boldsymbol{\theta}_I)\right)$
    **if** stop criterion is met **then**
        **break**
    **end if**
**end for**
```

5.1 Stop Criteria

The GAMP iteration in Equations (2.1)-(2.12) is to be iterated *until convergence*. However, in a practical setup one may terminate the iteration once the algorithm is *sufficiently close to convergence*. The challenge is then to find some criterion that describes when the algorithm has (almost) converged. Here we discuss a few such stop criteria that may be used with Algorithms 1-3. For a more general introduction to stop criteria for iterative methods see e.g. [46].

5.1.1 Mean Squared Error Stop Criterion

Since the GAMP iteration converges to a fixed point [25], it seems natural to stop the iteration once the change in the solutions between iterations becomes sufficiently small. If the change in the solution between iterations is measured in the 2-norm, we have a mean squared error (MSE) stop criterion

$$\frac{1}{n}||\bar{\boldsymbol{\alpha}}_{t\text{-}1} - \bar{\boldsymbol{\alpha}}_t||_2^2 < \epsilon \tag{5.1}$$

for some tolerance ϵ.

Note that if the algorithm stalls for some iterations, too early termination may occur when using this MSE stop criterion. It is the authors' experience that this may indeed happen with GAMP in some cases.

5.1.2 Normalised Mean Squared Error Stop Criterion

A stop criterion related to the MSE stop criterion in Equation (5.1) is the normalised mean squared error (NMSE) stop criterion used in e.g. [28], [31], [40]

$$\frac{||\bar{\boldsymbol{\alpha}}_{t\text{-}1} - \bar{\boldsymbol{\alpha}}_t||_2^2}{||\bar{\boldsymbol{\alpha}}_{t\text{-}1}||_2^2} < \epsilon \tag{5.2}$$

for some tolerance ϵ.

Note that this criterion is subject to a potential division by zero problem if $\bar{\boldsymbol{\alpha}}_{t\text{-}1} = \mathbf{0}$ which happens if the solution vector is initialised to the zero-vector. It is the authors' experience that this stop criterion is more robust towards stalls than the MSE criterion in Equation (5.1).

5.1.3 Residual Stop Criterion

If an additive measurement noise is assumed, as e.g. when using the AWGN GAMP output channel, one may define a stop criterion based on noise power. The GAMP iteration should then be terminated once the residual may be regarded as a reflection of the noise, i.e. once it has a signal power smaller than the noise power. Thus, we have the residual stop criterion

$$\frac{1}{m}||\mathbf{y} - \mathbf{A}\bar{\boldsymbol{\alpha}}_t||_2^2 < \epsilon \tag{5.3}$$

for some tolerance ϵ reflecting the noise power, e.g. $\epsilon = \sigma^2$ for an AWGN with variance σ^2.

5.1.4 Residual Measurements Ratio Stop Criterion

In [47] it is suggested to use the following residual measurments ratio stop criterion in iterative reconstruction methods

$$\frac{||\mathbf{y} - \mathbf{A}\bar{\boldsymbol{\alpha}}_t||_2^2}{||\mathbf{y}||_2^2} < \epsilon \tag{5.4}$$

for some tolerance ϵ.

Note that if the initial solution vector is chosen to be the zero-vector, this stop criterion expresses the ratio of the residual at iteration t to the residual at iteration one. Thus, convergence is determined based on the reduction in the residual. Also note that such a residual ratio stop criterion reflects the error in the solution through the condition number of the system matrix (for a non-singular system matrix) as detailed in [46].

5.2 Damping and Other Methods for Improving Convergence

The MMSE GAMP in Algorithm 2 incorporates damping of the GAMP updates by virtue of κ. It has been shown that the application of sufficient damping guarantees convergence of GAMP for arbitrary system matrices, $\mathbf{A}$, and with a Gaussian distributed vector, $\boldsymbol{\alpha}$, as well as for some other distributions on the vector $\boldsymbol{\alpha}$ under certain conditions [20]. Note that the GAMP states that are damped in [20] are slightly different from those that are damped in Algorithm 2.

An adaptive damping scheme is proposed in [31] along with a scheme for removing any non-zero mean of the system matrix. Such non-zero mean system matrices may significantly impede convergence of the GAMP algorithm [31], [48]. Another method for improving the convergence of the GAMP algorithm is to use a sequential updating scheme [48] where each of the elements in the GAMP states are updated one at a time instead of in parallel. This is the idea used in the Swept AMP from [49].

6 Parameter Learning

For the GAMP algorithm to converge, it is essential to use some learning or estimation scheme to update the AWGN output channel noise level (when using the AWGN output channel)[1]. Furthermore, several studies have shown that Expectation Maximization (EM) may be used to effectively learn input channel parameters [6], [28], [39], [45], [50] to the point where oracle-like performance is achieved. An alternative adaptive GAMP strategy for learning the channel distributions is detailed in [51] and [52].

6.1 Variance Estimates

For the AWGN output channel given in Equations (3.57) and (3.59) with noise variance σ^2, one may use a per iteration estimate of the noise variance. This may e.g. be done using the sample variance estimator

$$(\sigma^2)^{t+1} = \frac{1}{m}||\mathbf{y} - \mathbf{A}\bar{\boldsymbol{\alpha}}^{t+1}||_2^2 \tag{6.1}$$

Another option is to use the median based estimator often preferred by Donoho and Montanari [9], [47], [53]:

$$(\sigma^2)^{t+1} = \left(\text{median}\left(\frac{|\mathbf{y} - \mathbf{A}\bar{\boldsymbol{\alpha}}^{t+1}|}{\Phi_{\mathcal{N}}^{-1}(0.75)}\right)\right)^2 \tag{6.2}$$

In Equation (6.2), the absolute value is entrywise and $\Phi_{\mathcal{N}}^{-1}$ is the inverse standard normal cumulative distribution function.

6.2 Expectation Maximization (EM)

The Expectation Maximization (EM) algorithm may be used to find maximum likelihood (ML) estimates of parameters in probabilistic models [54] (see also [55] for an introduction to EM). Here we give an introduction to the use of EM to learn GAMP channel parameters as presented in [28], [39], [45].

The complete vector of GAMP channel parameters $\boldsymbol{\theta}_C$ is the concatenation of the input channel parameter vector(/matrix) $\boldsymbol{\theta}_I$ and the output channel parameter vector(/matrix) $\boldsymbol{\theta}_o$, i.e. $\boldsymbol{\theta}_C = [\boldsymbol{\theta}_I^T, \boldsymbol{\theta}_o^T]^T$. Now, the goal in using EM is to maximise the likelihood $p(\mathbf{y}|\boldsymbol{\theta}_C)$ with respect to $\boldsymbol{\theta}_C$. This is done using an iterative scheme in which each iteration has an E-step and an M-step that is guaranteed to increase the likelihood (if not at a stationary point already). One may elect to use a partial E-step in an "incremental" EM scheme [56] for improved convergence and/or to obtain a more computationally tractable problem. A partial M-step (known as the expectation conditional maximisation (ECM) algorithm [57]) is also an option to obtain a more computationally tractable problem. Both of these partial schemes are also guaranteed to increase the likelihood (if not at a stationary point already), though not necessarily maximise it, in each iteration.

For the general GAMP channel parameter optimisation problem, the EM algorithm manifests as the recursion of the following optimisation problem[2] [28]

$$\boldsymbol{\theta}_C^{t+1} = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{\alpha}|\mathbf{y},\boldsymbol{\theta}_C^t}[\ln(p(\mathbf{y}, \boldsymbol{\alpha}; \boldsymbol{\theta}))] \tag{6.3}$$

where $\mathbf{y}$ is the vector of observed variables (the measurements) and $\boldsymbol{\alpha}$ is the vector of the latent (unobserved or hidden) variables. Specifically, $\boldsymbol{\alpha}$ is the coefficient vector in Equation (1.1). Now,

[1] At least that is what the authors of this tech report have experienced in an extensive set of simulations

[2] Strictly speaking, this is the M-step in the EM algorithm consisting of both an E-step and the M-step. However, the E-step amounts to trivially choosing the distribution $p(\boldsymbol{\alpha}|\mathbf{y}; \boldsymbol{\theta}_C^t)$ for the expectation [28].

if we consider only updating the input channel parameters (a partial M-step approach), we have

$$
\begin{aligned}
\boldsymbol{\theta}_I^{t+1} &= \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{\alpha}|\mathbf{y},\boldsymbol{\theta}_C^t}[\ln(p(\mathbf{y},\boldsymbol{\alpha};\boldsymbol{\theta}))] && (6.4)\\
&= \arg\max_{\boldsymbol{\theta}} \int_{\boldsymbol{\alpha}} p(\boldsymbol{\alpha}|\mathbf{y};\boldsymbol{\theta}_C^t)\ln(p(\mathbf{y},\boldsymbol{\alpha};\boldsymbol{\theta}))d\boldsymbol{\alpha} && (6.5)\\
&= \arg\max_{\boldsymbol{\theta}} \int_{\boldsymbol{\alpha}} p(\boldsymbol{\alpha}|\mathbf{y};\boldsymbol{\theta}_C^t)\ln(p(\mathbf{y}|\boldsymbol{\alpha};\boldsymbol{\theta}_o)p(\boldsymbol{\alpha};\boldsymbol{\theta}))d\boldsymbol{\alpha} && (6.6)\\
&= \arg\max_{\boldsymbol{\theta}} \left(\int_{\boldsymbol{\alpha}} p(\boldsymbol{\alpha}|\mathbf{y};\boldsymbol{\theta}_C^t)\ln(p(\mathbf{y}|\boldsymbol{\alpha};\boldsymbol{\theta}_o))d\alpha + \int_{\boldsymbol{\alpha}} p(\boldsymbol{\alpha}|\mathbf{y};\boldsymbol{\theta}_C^t)\ln(p(\boldsymbol{\alpha};\boldsymbol{\theta}))d\boldsymbol{\alpha}\right) && (6.7)\\
&= \arg\max_{\boldsymbol{\theta}} \int_{\boldsymbol{\alpha}} p(\boldsymbol{\alpha}|\mathbf{y};\boldsymbol{\theta}_C^t)\ln(p(\boldsymbol{\alpha};\boldsymbol{\theta}))d\boldsymbol{\alpha} + const. && (6.8)\\
&= \arg\max_{\boldsymbol{\theta}} \int_{\boldsymbol{\alpha}} p(\boldsymbol{\alpha}|\mathbf{y};\boldsymbol{\theta}_C^t)\ln(p(\boldsymbol{\alpha};\boldsymbol{\theta}))d\boldsymbol{\alpha} && (6.9)\\
&= \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{\alpha}|\mathbf{y},\boldsymbol{\theta}_C^t}[\ln(p(\boldsymbol{\alpha};\boldsymbol{\theta}))] && (6.10)
\end{aligned}
$$

since, by definition, $\boldsymbol{\theta}_I$ is only used in the specification of the prior $p(\boldsymbol{\alpha};\boldsymbol{\theta}_I)$. That is, given a specific value of $\boldsymbol{\alpha}$, the value of $\mathbf{y}$ no longer depends on $\boldsymbol{\theta}_I$. Thus, w.r.t. elements of $\boldsymbol{\theta}_I$, $p(\mathbf{y}|\boldsymbol{\alpha};\boldsymbol{\theta}_o)$ is a constant.

In order to find a similar expression for a separate update of the output channel parameters (i.e. a partial M-step approach update of $\boldsymbol{\theta}_o$), one must realise that the distribution of $\mathbf{y}$ depends only on $\boldsymbol{\alpha}$ through $\mathbf{z}$ since $\mathbf{z} = \mathbf{A}\boldsymbol{\alpha}$ (a deterministic relation) as specified in Equation (1.6). Due to the separability assumption of GAMP (see the discussion below Equations (1.16) and (1.17)), we then have a Markov chain $\boldsymbol{\alpha} \to \mathbf{z} \to \mathbf{y}$ meaning that $p(\boldsymbol{\alpha},\mathbf{y}|\mathbf{z}) = p(\boldsymbol{\alpha}|\mathbf{z})p(\mathbf{y}|\mathbf{z})$. The deterministic relation $\mathbf{z} = \mathbf{A}\boldsymbol{\alpha}$ also means that the joint distribution $p(\boldsymbol{\alpha},\mathbf{z})$ is degenerate[3]. All of this makes for a series of mathematical subtleties in the below expression. However, in accepting these expressions, it is probably most important to realise that *the distribution of* $\mathbf{y}$ *depends only on* $\boldsymbol{\alpha}$ *through* $\mathbf{z}$.

$$
\begin{aligned}
\boldsymbol{\theta}_o^{t+1} &= \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{\alpha}|\mathbf{y},\boldsymbol{\theta}_C^t}[\ln(p(\mathbf{y},\boldsymbol{\alpha};\boldsymbol{\theta}))] && (6.11)\\
&= \arg\max_{\boldsymbol{\theta}} \int_{\boldsymbol{\alpha}} p(\boldsymbol{\alpha}|\mathbf{y};\boldsymbol{\theta}_C^t)\ln(p(\mathbf{y},\boldsymbol{\alpha};\boldsymbol{\theta}))d\boldsymbol{\alpha} && (6.12)\\
&= \arg\max_{\boldsymbol{\theta}} \int_{\boldsymbol{\alpha}} p(\boldsymbol{\alpha}|\mathbf{y};\boldsymbol{\theta}_C^t)\ln(p(\mathbf{y}|\boldsymbol{\alpha};\boldsymbol{\theta})p(\boldsymbol{\alpha};\boldsymbol{\theta}_I))d\boldsymbol{\alpha} && (6.13)\\
&= \arg\max_{\boldsymbol{\theta}} \left(\int_{\boldsymbol{\alpha}} p(\boldsymbol{\alpha}|\mathbf{y};\boldsymbol{\theta}_C^t)\ln(p(\mathbf{y}|\boldsymbol{\alpha};\boldsymbol{\theta}))d\alpha + \int_{\boldsymbol{\alpha}} p(\boldsymbol{\alpha}|\mathbf{y};\boldsymbol{\theta}_C^t)\ln(p(\boldsymbol{\alpha};\boldsymbol{\theta}_I))d\boldsymbol{\alpha}\right) && (6.14)\\
&= \arg\max_{\boldsymbol{\theta}} \int_{\boldsymbol{\alpha}} p(\boldsymbol{\alpha}|\mathbf{y};\boldsymbol{\theta}_C^t)\ln(p(\mathbf{y}|\boldsymbol{\alpha};\boldsymbol{\theta}))d\boldsymbol{\alpha} + const. && (6.15)\\
&= \arg\max_{\boldsymbol{\theta}} \int_{\boldsymbol{\alpha}} \frac{p(\boldsymbol{\alpha},\mathbf{y};\boldsymbol{\theta}_C^t)}{p(\mathbf{y};\boldsymbol{\theta}_C^t)}\ln(p(\mathbf{y}|\boldsymbol{\alpha};\boldsymbol{\theta}))d\boldsymbol{\alpha} + const. && (6.16)\\
&= \arg\max_{\boldsymbol{\theta}} \int_{\boldsymbol{\alpha}}\int_{\mathbf{z}} \frac{p(\boldsymbol{\alpha},\mathbf{y},\mathbf{z};\boldsymbol{\theta}_C^t)}{p(\mathbf{y};\boldsymbol{\theta}_C^t)}d\mathbf{z}\ln(p(\mathbf{y}|\boldsymbol{\alpha};\boldsymbol{\theta}))d\boldsymbol{\alpha} + const. && (6.17)\\
&= \arg\max_{\boldsymbol{\theta}} \int_{\boldsymbol{\alpha}}\int_{\mathbf{z}} \frac{p(\boldsymbol{\alpha},\mathbf{y}|\mathbf{z};\boldsymbol{\theta}_C^t)p(\mathbf{z};\boldsymbol{\theta}_C^t)}{p(\mathbf{y};\boldsymbol{\theta}_C^t)}d\mathbf{z}\ln(p(\mathbf{y}|\boldsymbol{\alpha};\boldsymbol{\theta}))d\boldsymbol{\alpha} + const. && (6.18)\\
&= \arg\max_{\boldsymbol{\theta}} \int_{\boldsymbol{\alpha}}\int_{\mathbf{z}} \frac{p(\boldsymbol{\alpha}|\mathbf{z};\boldsymbol{\theta}_C^t)p(\mathbf{y}|\mathbf{z};\boldsymbol{\theta}_C^t)p(\mathbf{z};\boldsymbol{\theta}_C^t)}{p(\mathbf{y};\boldsymbol{\theta}_C^t)}d\mathbf{z}\ln(p(\mathbf{y}|\boldsymbol{\alpha};\boldsymbol{\theta}))d\boldsymbol{\alpha} + const. && (6.19)
\end{aligned}
$$

[3] To see this, consider the simple example that $\boldsymbol{\alpha} = [\alpha_1, \alpha_2]^T$ with $p(\alpha_1,\alpha_2) = p(\alpha_1)p(\alpha_2)$, $p(\alpha_1) = \mathcal{N}(\alpha_1; 0, 1)$, $p(\alpha_2) = \mathcal{N}(\alpha_2; 0, 1)$, and $z = \alpha_1 + \alpha_2$. Now, $p(z)$ is well defined since it is simply the convolution of $p(\alpha_1)$ and $p(\alpha_2)$. However, $p(\boldsymbol{\alpha}, z)$ is degenerate in the sense the "density" is restricted to values for which $z = \alpha_1 + \alpha_2$. Similarly, $p(\boldsymbol{\alpha}|z) = \frac{p(\boldsymbol{\alpha},z)}{p(z)}$ as well as $p(z|\boldsymbol{\alpha}) = \frac{p(\boldsymbol{\alpha},z)}{p(\boldsymbol{\alpha})}$ are degenerate for the same reason. Thus, they act as a sort of sampling that squeezes the domain of $\boldsymbol{\alpha}$ into the domain of z since z is deterministically derived from $\boldsymbol{\alpha}$. In a sense they have a "sampling property" similar to that of the generalised Dirac delta function.

$$= \arg\max_{\boldsymbol{\theta}} \int_{\boldsymbol{\alpha}} \int_{\mathbf{z}} p(\boldsymbol{\alpha}|\mathbf{z};\boldsymbol{\theta}_C^t) p(\mathbf{z}|\mathbf{y};\boldsymbol{\theta}_C^t) d\mathbf{z} \ln(p(\mathbf{y}|\boldsymbol{\alpha};\boldsymbol{\theta})) d\boldsymbol{\alpha} + const. \tag{6.20}$$

$$= \arg\max_{\boldsymbol{\theta}} \int_{\mathbf{z}} p(\mathbf{z}|\mathbf{y};\boldsymbol{\theta}_C^t) \int_{\boldsymbol{\alpha}} p(\boldsymbol{\alpha}|\mathbf{z};\boldsymbol{\theta}_C^t) \ln(p(\mathbf{y}|\boldsymbol{\alpha};\boldsymbol{\theta})) d\boldsymbol{\alpha} d\mathbf{z} + const. \tag{6.21}$$

$$= \arg\max_{\boldsymbol{\theta}} \int_{\mathbf{z}} p(\mathbf{z}|\mathbf{y};\boldsymbol{\theta}_C^t) \ln(p(\mathbf{y}|\mathbf{z};\boldsymbol{\theta})) d\mathbf{z} + const. \tag{6.22}$$

$$= \arg\max_{\boldsymbol{\theta}} \int_{\mathbf{z}} p(\mathbf{z}|\mathbf{y};\boldsymbol{\theta}_C^t) \ln(p(\mathbf{y}|\mathbf{z};\boldsymbol{\theta})) d\mathbf{z} \tag{6.23}$$

$$= \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{z}|\mathbf{y},\boldsymbol{\theta}_C^t}[\ln(p(\mathbf{y}|\mathbf{z};\boldsymbol{\theta}))] \tag{6.24}$$

where the constant is due to $p(\boldsymbol{\alpha};\boldsymbol{\theta}_I)$ being constant w.r.t. elements of $\boldsymbol{\theta}_o$.

Now since we are using GAMP which is trying to (indirectly) find the true posteriors $p(\boldsymbol{\alpha}|\mathbf{y},\boldsymbol{\theta}_C)$ and $p(\mathbf{z}|\mathbf{y},\boldsymbol{\theta}_C)$, we only have the GAMP approximated posteriors in Equations (3.4) and (3.8), respectively, available. Thus, these GAMP approximations are used in computing the expectation in the EM update [28], i.e., the E-step becomes approximate[4]. We then have the final GAMP EM channel parameter recursions

$$\boldsymbol{\theta}_I^{t+1} = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{\alpha}|\mathbf{y},\mathbf{s},\mathbf{r},\boldsymbol{\theta}_I^t}[\ln(p(\boldsymbol{\alpha};\boldsymbol{\theta}))] \tag{6.25}$$

$$= \arg\max_{\boldsymbol{\theta}} \sum_{j=1}^{n} \mathbb{E}_{\boldsymbol{\alpha}|\mathbf{y},\mathbf{s},\mathbf{r},\boldsymbol{\theta}_I^t}[\ln(p(\alpha_j;[\boldsymbol{\theta}]_j))] \tag{6.26}$$

$$\boldsymbol{\theta}_o^{t+1} = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{z}|\mathbf{y},\mathbf{v},\mathbf{o},\boldsymbol{\theta}_o^t}[\ln(p(\mathbf{y}|\mathbf{z};\boldsymbol{\theta}))] \tag{6.27}$$

$$= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{m} \mathbb{E}_{\mathbf{z}|\mathbf{y},\mathbf{v},\mathbf{o},\boldsymbol{\theta}_o^t}[\ln(p(y_i|z_i;[\boldsymbol{\theta}]_i))] \tag{6.28}$$

where we have used the separability properties of the in- and output channels as described in Equations (1.16) and (1.17). Note that the GAMP approximated posteriors $p(\mathbf{z}|\mathbf{y},\mathbf{v},\mathbf{o},\boldsymbol{\theta}_o^t)$ and $p(\boldsymbol{\alpha}|\mathbf{y},\mathbf{s},\mathbf{r},\boldsymbol{\theta}_I^t)$ by definition are separable (as everything else in the in- and output channels). Also note that one may use several GAMP iterations to find $p(\boldsymbol{\alpha}|\mathbf{y},\mathbf{s},\mathbf{r},\boldsymbol{\theta}_I^t)$ and $p(\mathbf{z}|\mathbf{y},\mathbf{v},\mathbf{o},\boldsymbol{\theta}_o^t)$ for each EM update in Equations (6.26), (6.28) [28].

Furthermore, Schniter and Vila [28] as well as Krzakala et al. [5] use a "complete" partial M-step in the sense that the elements of $\boldsymbol{\theta}_C$ are updated one at a time, i.e. they essentially use the expectation/conditional maximization (ECM) algorithm [57]. Thus, in the following, our focus is on finding recursions based on Equations (6.26) and (6.28) for one parameter (one element of $\boldsymbol{\theta}_C$) at a time. When using such a scheme, the ordering of the updates of the elements of $\boldsymbol{\theta}_C$ become important since all parameter updates should be based on the latest value of all other parameters. The particular choice of update order may be arbitrary, but the all updates must be based on the most recent values of all other parameters that they depend on. Note, though, that Schniter and Vila [28] use the GAMP approximated posteriors $p(\mathbf{z}|\mathbf{y},\mathbf{v},\mathbf{o},\boldsymbol{\theta}_o^t)$ and $p(\boldsymbol{\alpha}|\mathbf{y},\mathbf{s},\mathbf{r},\boldsymbol{\theta}_I^t)$ based on iteration t for the all parameter updates. That is, they do not recompute the GAMP posteriors between the parameter updates.

[4] It is not clear whether or not this falls under the framework of partial E-steps from [56] guaranteed to increase the likelihood.

6.2.1 EM Updates for Common Output Channels

We now state EM updates for several commonly used GAMP output channels.

6.2.1.1 AWGN Output Channel

For the AWGN output channel given in Equations (3.57) and (3.59) with noise variance σ^2, Krzakala suggested the following EM recursion on the noise variance [5] (Eq. 77)

$$(\sigma^2)^{t+1} = \frac{\sum_i \frac{(y_i - o_i^{t+1})^2}{\left(1 + \frac{1}{(\sigma^2)^t} v_i^{t+1}\right)^2}}{\sum_i \frac{1}{1 + \frac{1}{(\sigma^2)^t} v_i^{t+1}}} \tag{6.29}$$

Manoel and Tramel suggested the following implementation[5] of the EM recursion in Equation (6.29):

$$(\sigma^2)^{t+1} = (\sigma^2)^t \frac{\sum_i \left(\frac{y_i - o_i^{t+1}}{(\sigma^2)^t + v_i^{t+1}}\right)^2}{\sum_i ((\sigma^2)^t + v_i^{t+1})^{-1}} \tag{6.30}$$

which in matrix-vector + Numpy broadcast notation is:

$$\sigma_t^2 = \sigma_{t\text{-}1}^2 \frac{\sum \left(\frac{\mathbf{y} - \mathbf{o}_t}{\sigma_{t\text{-}1}^2 + \mathbf{v}_t}\right)}{\sum (\sigma_{t\text{-}1}^2 + \mathbf{v}_t)^{-1}} \tag{6.31}$$

Note that $\mathbf{v}_t$ becomes scalar when using the sum approximation GAMP described in Section 4 and Algorithm 3. Thus, one must make sure to implement the denominator sum in Equation (6.31) such that it acts as if $\mathbf{v}_t$ was a vector (a sum over m elements - not just a single element).

Schniter and Vila suggested the following EM recursion on the noise variance [28] (Eq. (27)), [39] (Eq. (38))

$$(\sigma^2)^{t+1} = \frac{1}{m} \sum_i (|y_i - \bar{z}_i^{t+1}|^2 + \tilde{z}_i^{t+1}) \tag{6.32}$$

However, in [40] it is reported that the closely related expression

$$(\sigma^2)^{t+1} = \frac{1}{m} (||\mathbf{y} - \mathbf{A}\bar{\boldsymbol{\alpha}}^{t+1}||_2^2 + \sum_i [|\mathbf{A}|^{\circ 2} \tilde{\boldsymbol{\alpha}}^{t+1}]_i) \tag{6.33}$$

is supposed to yield improved performance in low SNR cases (SNR $<$ 10 dB). Note how this expression is an extension of Equation (6.1).

6.2.1.2 AWLN Output Channel

For the AWLN output channel given in Equations (3.62) and (3.63) with rate parameter λ, Vila and Schniter suggested the following EM recursion on the rate parameter [40] (Eq. (52))

$$\lambda^{t+1} = \frac{m}{\sum_i \left(\Phi_{\mathcal{N}}\left(\frac{-\hat{z}_i}{v_i}\right)\left(\hat{z}_i + \sqrt{v_i}\frac{\phi_{\mathcal{N}}\left(\frac{\hat{z}_i}{\sqrt{v_i}}\right)}{\Phi_{\mathcal{N}}\left(\frac{\hat{z}_i}{\sqrt{v_i}}\right)}\right) - \Phi_{\mathcal{N}}\left(\frac{\hat{z}_i}{v_i}\right)\left(\hat{z}_i - \sqrt{v_i}\frac{\phi_{\mathcal{N}}\left(\frac{-\hat{z}_i}{\sqrt{v_i}}\right)}{\Phi_{\mathcal{N}}\left(\frac{-\hat{z}_i}{\sqrt{v_i}}\right)}\right)\right)} \tag{6.34}$$

for

$$\hat{\mathbf{z}} \coloneqq \mathbf{A}\boldsymbol{\alpha} - \mathbf{y} \tag{6.35}$$

[5] See: `https://github.com/eric-tramel/SwAMP-Demo/blob/master/python/amp.py`

6.2.2 EM Updates for Common Input Channels

We now state EM updates for several commonly used GAMP input channels.

6.2.2.1 General i.i.d. Sparse Input Channel

As noted in [28], one may find a general expression for the EM update of the signal density τ in the general sparse i.i.d. input prior in Equation (3.29). In particular, we have

$$\begin{aligned}\tau^{t+1} &= \operatorname*{arg\,max}_{\tau\in[0,1]} \sum_{j=1}^{n} \mathbb{E}_{\boldsymbol{\alpha}|\mathbf{y},\mathbf{s},\mathbf{r},\boldsymbol{\theta}_I^t}[\ln\left(p(\alpha_j;\tau,\boldsymbol{\theta}_{I\setminus\tau}^t)\right)] && (6.36)\\ &= \operatorname*{arg\,max}_{\tau\in[0,1]} \sum_{j=1}^{n} \int_{\alpha_j} p(\alpha_j|\mathbf{y},s_j,r_j,\boldsymbol{\theta}_I^t)\ln\left((1-\tau)\delta_{\text{Dirac}}(\alpha_j)+\tau\varphi(\alpha_j;\boldsymbol{\theta}_{I\setminus\tau})\right)d\alpha_j && (6.37)\end{aligned}$$

Setting the derivative of the objective equal to zero, we obtain

$$\begin{aligned}0 &= \frac{\partial}{\partial\tau}\sum_{j=1}^{n}\int_{\alpha_j} p(\alpha_j|\mathbf{y},s_j,r_j,\boldsymbol{\theta}_I^t)\ln\left((1-\tau)\delta_{\text{Dirac}}(\alpha_j)+\tau\varphi(\alpha_j;\boldsymbol{\theta}_{I\setminus\tau})\right)d\alpha_j && (6.38)\\ &= \sum_{j=1}^{n}\int_{\alpha_j} p(\alpha_j|\mathbf{y},s_j,r_j,\boldsymbol{\theta}_I^t)\frac{\partial}{\partial\tau}\ln\left((1-\tau)\delta_{\text{Dirac}}(\alpha_j)+\tau\varphi(\alpha_j;\boldsymbol{\theta}_{I\setminus\tau})\right)d\alpha_j && (6.39)\end{aligned}$$

where we have used Leibniz's integral rule to exchange differentiation and integration. Leibniz's integral rule requires the integrand and its partial derivative w.r.t. τ to be continuous in both α_j and τ which is not strictly true for the above objective due to the Dirac delta function. However, as noted in [40], one may justify its use by considering an approximation of the Dirac delta function $\delta_{\text{Dirac}}(\alpha)\approx\mathcal{N}(\alpha,0,\epsilon)$ for a fixed arbitrarily small $\epsilon>0$. For the partial derivative, we have

$$\begin{aligned}\frac{\partial}{\partial\tau}\ln\left((1-\tau)\delta_{\text{Dirac}}(\alpha_j)+\tau\varphi(\alpha_j;\boldsymbol{\theta}_{I\setminus\tau})\right) &= \frac{\varphi(\alpha_j;\boldsymbol{\theta}_{I\setminus\tau})-\delta_{\text{Dirac}}(\alpha_j)}{(1-\tau)\delta_{\text{Dirac}}(\alpha_j)+\tau\varphi(\alpha_j;\boldsymbol{\theta}_{I\setminus\tau})} && (6.40)\\ &= \frac{\frac{\varphi(\alpha_j;\boldsymbol{\theta}_{I\setminus\tau})}{\delta_{\text{Dirac}}(\alpha_j)}-1}{(1-\tau)+\tau\frac{\varphi(\alpha_j;\boldsymbol{\theta}_{I\setminus\tau})}{\delta_{\text{Dirac}}(\alpha_j)}} && (6.41)\\ &= \begin{cases}\frac{-1}{1-\tau}, & \alpha_j=0\\ \frac{1}{\tau}, & \alpha_j\neq 0\end{cases} && (6.42)\end{aligned}$$

Following [28], we may define a closed ball $\mathcal{B}_\epsilon := [-\epsilon,\epsilon]$ and its complement $\overline{\mathcal{B}}_\epsilon = \mathbb{R}\setminus\mathcal{B}_\epsilon$ which may be used to evaluate the integral in Equation (6.39) as $\epsilon\to 0$. Then using the GAMP approximated posterior in Equation (3.37), we get

$$\begin{aligned}0 &= \sum_{j=1}^{n}\left(\frac{-1}{1-\tau}\int_{\alpha_j\in\mathcal{B}_\epsilon} p(\alpha_j|\mathbf{y},s_j,r_j,\boldsymbol{\theta}_I^t)d\alpha_j + \frac{1}{\tau}\int_{\alpha_j\in\overline{\mathcal{B}}_\epsilon} p(\alpha_j|\mathbf{y},s_j,r_j,\boldsymbol{\theta}_I^t)d\alpha_j\right) && (6.43)\\ &= \sum_{j=1}^{n}\Bigg(\frac{-1}{1-\tau}(1-\pi(r_j,s_j,\boldsymbol{\theta}_I^t))\int_{\alpha_j\in\mathcal{B}_\epsilon}\delta_{\text{Dirac}}(\alpha_j)d\alpha_j+\\ &\qquad \frac{1}{\tau}\pi(r_j,s_j,\boldsymbol{\theta}_I^t)\int_{\alpha_j\in\overline{\mathcal{B}}_\epsilon}\varphi_{\alpha_j|\mathbf{y};s_j,r_j,\boldsymbol{\theta}_I^t}(\alpha_j;\boldsymbol{\theta}_I^t)d\alpha_j\Bigg) && (6.44)\\ &= \sum_{j=1}^{n}\left(\frac{-1}{1-\tau}(1-\pi(r_j,s_j,\boldsymbol{\theta}_I^t))+\frac{1}{\tau}\pi(r_j,s_j,\boldsymbol{\theta}_I^t)\right) && (6.45)\\ &= \sum_{j=1}^{n}\left(\frac{-\tau}{1-\tau}(1-\pi(r_j,s_j,\boldsymbol{\theta}_I^t))+\pi(r_j,s_j,\boldsymbol{\theta}_I^t)\right) && (6.46)\end{aligned}$$

$$= \frac{-\tau n}{1-\tau} - \frac{-\tau}{1-\tau}\sum_{j=1}^{n}\pi(r_j, s_j, \boldsymbol{\theta}_I^t) + \sum_{j=1}^{n}\pi(r_j, s_j, \boldsymbol{\theta}_I^t) \tag{6.47}$$

$$= -\tau n + \tau\sum_{j=1}^{n}\pi(r_j, s_j, \boldsymbol{\theta}_I^t) + \sum_{j=1}^{n}\pi(r_j, s_j, \boldsymbol{\theta}_I^t) - \tau\sum_{j=1}^{n}\pi(r_j, s_j, \boldsymbol{\theta}_I^t) \tag{6.48}$$

$$= -n\tau + \sum_{j=1}^{n}\pi(r_j, s_j, \boldsymbol{\theta}_I^t) \tag{6.49}$$

with $\pi(r_j, s_j, \boldsymbol{\theta}_I^t)$ defined in Equation (3.41). Note that in going from Equation (6.44) to Equation (6.45), we have assumed that $\varphi(\alpha_j; \boldsymbol{\theta}_I^t)$ is well behaved at $\alpha_j = 0$ such that

$$\int_{\alpha_j \in \overline{\mathcal{B}_\epsilon}} \varphi_{\alpha_j|\mathbf{y};s_j,r_j,\boldsymbol{\theta}_I^t}(\alpha_j; \boldsymbol{\theta}_I^t) d\alpha_j = \int_{\alpha_j} \varphi_{\alpha_j|\mathbf{y};s_j,r_j,\boldsymbol{\theta}_I^t}(\alpha_j; \boldsymbol{\theta}_I^t) d\alpha_j = 1 \tag{6.50}$$

Solving for τ, we get the final expression for its EM update

$$\tau^{t+1} = \frac{1}{n}\sum_{j=1}^{n}\pi(r_j, s_j, \boldsymbol{\theta}_I^t) \tag{6.51}$$

The update in Equation (6.51) is intuitively pleasing since it states that the signal density τ is the average of the posterior signal density (posterior support probabilities) $\pi(r_j, s_j, \boldsymbol{\theta}_I^t)$. Also note that since $\pi(r_j, s_j, \boldsymbol{\theta}_I^t) \in [0;1], \forall j$, we have that $\tau^{t+1} \in [0;1]$.

6.2.2.2 General Weighted Sparse Input Channel

An EM update of the common signal density τ in the GWS input prior in Equation (3.44) may be found using similar derivations as those used for the general i.i.d. sparse input channel detailed in Section 6.2.2.1. In particular, we have

$$\tau^{t+1} = \operatorname*{arg\,max}_{\tau\in[0,1]} \sum_{j=1}^{n} \mathbb{E}_{\boldsymbol{\alpha}|\mathbf{y},\mathbf{s},\mathbf{r},\boldsymbol{\theta}_I^t}[\ln\left(p(\alpha_j; \tau, \boldsymbol{\theta}_{I\setminus\tau}^t)\right)] \tag{6.52}$$

$$= \operatorname*{arg\,max}_{\tau\in[0,1]} \sum_{j=1}^{n} \int_{\alpha_j} p(\alpha_j|\mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t) \ln\left((1-w_j\tau)\delta_{\text{Dirac}}(\alpha_j) + w_j\tau\varphi(\alpha_j; \boldsymbol{\theta}_{I\setminus\tau})\right) d\alpha_j \tag{6.53}$$

Setting the derivative of the objective equal to zero, we obtain

$$0 = \frac{\partial}{\partial\tau}\sum_{j=1}^{n}\int_{\alpha_j} p(\alpha_j|\mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t) \ln\left((1-w_j\tau)\delta_{\text{Dirac}}(\alpha_j) + w_j\tau\varphi(\alpha_j; \boldsymbol{\theta}_{I\setminus\tau})\right) d\alpha_j \tag{6.54}$$

$$= \sum_{j=1}^{n}\int_{\alpha_j} p(\alpha_j|\mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t) \frac{\partial}{\partial\tau} \ln\left((1-w_j\tau)\delta_{\text{Dirac}}(\alpha_j) + w_j\tau\varphi(\alpha_j; \boldsymbol{\theta}_{I\setminus\tau})\right) d\alpha_j \tag{6.55}$$

where we again have used Leibniz's integral rule to exchange differentiation and integration. For the partial derivative, we have

$$\frac{\partial}{\partial\tau}\ln\left((1-w_j\tau)\delta_{\text{Dirac}}(\alpha_j) + w_j\tau\varphi(\alpha_j; \boldsymbol{\theta}_{I\setminus\tau})\right)$$

$$= \frac{w_j\varphi(\alpha_j; \boldsymbol{\theta}_{I\setminus\tau}) - w_j\delta_{\text{Dirac}}(\alpha_j)}{(1-w_j\tau)\delta_{\text{Dirac}}(\alpha_j) + w_j\tau\varphi(\alpha_j; \boldsymbol{\theta}_{I\setminus\tau})} \tag{6.56}$$

$$= \frac{w_j\frac{\varphi(\alpha_j;\boldsymbol{\theta}_{I\setminus\tau})}{\delta_{\text{Dirac}}(\alpha_j)} - w_j}{(1-w_j\tau) + w_j\tau\frac{\varphi(\alpha_j;\boldsymbol{\theta}_{I\setminus\tau})}{\delta_{\text{Dirac}}(\alpha_j)}} \tag{6.57}$$

$$= \begin{cases} \frac{-w_j}{1-w_j\tau}, & \alpha_j = 0 \\ \frac{1}{\tau}, & \alpha_j \neq 0 \end{cases} \tag{6.58}$$

Again, using the closed ball $\mathcal{B}_\epsilon := [-\epsilon, \epsilon]$ and its complement $\overline{\mathcal{B}}_\epsilon = \mathbb{R} \setminus \mathcal{B}_\epsilon$ in evaluating the integral in Equation (6.55) as $\epsilon \to 0$ and using the GAMP approximated posterior in Equation (3.45), we get

$$0 = \sum_{j=1}^{n} \left(\frac{-w_j}{1 - w_j \tau} \int_{\alpha_j \in \mathcal{B}_\epsilon} p(\alpha_j | \mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t) d\alpha_j + \frac{1}{\tau} \int_{\alpha_j \in \overline{\mathcal{B}}_\epsilon} p(\alpha_j | \mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t) d\alpha_j \right) \tag{6.59}$$

$$= \sum_{j=1}^{n} \left(\frac{-w_j}{1 - w_j \tau} (1 - \pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t)) + \frac{1}{\tau} \pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t) \right) \tag{6.60}$$

$$= \sum_{j=1}^{n} \left(\frac{-w_j \tau}{1 - w_j \tau} (1 - \pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t)) + \pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t) \right) \tag{6.61}$$

$$= \sum_{j=1}^{n} \left(\frac{-w_j \tau}{1 - w_j \tau} - \frac{-w_j \tau}{1 - w_j \tau} \pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t) + \pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t) \right) \tag{6.62}$$

$$= \sum_{j=1}^{n} \left(-w_j \tau + w_j \tau \pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t) + \pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t) - w_j \tau \pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t) \right) \tag{6.63}$$

$$= -\tau \sum_{j=1}^{n} w_j + \sum_{j=1}^{n} \pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t) \tag{6.64}$$

with $\pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t)$ defined as in Equation (3.46). Solving for τ, we get the final expression for its EM update:

$$\tau^{t+1} = \frac{\sum_{j=1}^{n} \pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t)}{\sum_{j=1}^{n} w_j} \tag{6.65}$$

Note that the GWS τ EM update in Equation (6.65) reduces to the general sparse channel τ EM updates in Equation (6.51) for the choice of weights $\forall j, w_j = 1$.

In order for the GAMP updates to be stable, the GAMP posterior must remain a proper density which requires that $w_j \tau^{t+1} \in [0; 1]$, $\forall j$ as described in Section 3.5. If the requirement on the choice of weights is $w_j \in [0; 1]$, $\forall j$, then one must also require that $\tau^{t+1} \in [0; 1]$. Now, consider the case of having just a single element in the coefficient vector. The τ EM update then becomes

$$\tau^{t+1} = \frac{\pi_1^{\mathrm{w}}(r_1, s_1, \boldsymbol{\theta}_I^t)}{w_1} \in [0; \frac{1}{w_1}] \tag{6.66}$$

Thus, if the GAMP posterior support probability $\pi_1^{\mathrm{w}}(r_1, s_1, \boldsymbol{\theta}_I^t)$ is large and we have chosen a small w_1, we may end up with $\tau^{t+1} > 1$. That is, if there is a significant mismatch between our prior belief about the support probability (expressed by w_1 and the actual posterior support probability $\pi_1^{\mathrm{w}}(r_1, s_1, \boldsymbol{\theta}_I^t)$, it may potentially violate the requirement that $\tau^{t+1} \in [0; 1]$. In generalising this result to arbitrary length vectors we may consider

$$\tau^{t+1} = \frac{\sum_{j=1}^{n} \pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t)}{\sum_{j=1}^{n} w_j} = \frac{\frac{1}{n} \sum_{j=1}^{n} \pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t)}{\frac{1}{n} \sum_{j=1}^{n} w_j} \tag{6.67}$$

That is, if the average GAMP posterior support probability becomes larger than our prior average belief about the support probabilities (expressed by the w_j's), it may violate the requirement that $\tau^{t+1} \in [0; 1]$. Note that, as discussed in Section 6.2.2.1, this problem is not present if $w_j = 1$, $\forall j$.

At least two strategies for handling this violation can be identified

1. We may force $\tau^{t+1} = 1$ whenever $\tau^{t+1} > 1$. This may be interpreted as forcing the prior belief on the support probabilities. Note that since τ models the overall average sparsity of the signal, forcing $\tau \leq 1$ has the effect of forcing the average sparsity in the next GAMP iteration to be no larger than the average of the weights.

2. We may adjust the weights towards $w_j = 1$, $\forall j$ according to some scheme detailing the weights update. This strategy allows for the weighted model to adapt towards a non-weighted model, if the data suggest such a change.

One scheme for adjusting the weights towards $w_j = 1$, $\forall j$ is to apply Algorithm 4 which attempts to increase all weights such that $\tau = 1$. Worst case using this scheme, we get $w_j = 1$, $\forall j$ after n iterations of the while loop. In a practical implementation of Algorithm 4, it may be beneficial to introduce some other stop criterion.

Algorithm 4 - A scheme for adjusting the weights towards $w_j = 1$, $\forall j$ in the general weighted sparse GAMP input channel.

```
while $\tau > 1$ do
    for $j \in 1, \dots, n$ do
        $w_j = \tau w_j$
        if $w_j > 1$ then
            $w_j = 1$
        end if
    end for
    $\tau = \frac{\sum_{j=1}^{n} \pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t)}{\sum_{j=1}^{n} w_j}$
end while
```

6.2.2.3 EM updates of other channel parameters in the GWS channel

The derivation of EM updates for channel parameters in a general sparse input channel as detailed in [28], [39], [45] may be summarised as follows:

1. Pick a parameter to update.
2. Write down the single step EM-update as in e.g. Equation (6.53).
3. Take the partial derivative, set equal to zero, and apply Leibniz's integral rule to interchange integration and differentiation (assuming that this interchange is valid) as in e.g. Equation (6.55).
4. Compute the partial derivative as in e.g. Equation (6.58).
5. Handle the discontinuity at zero by splitting the integration and treating a closed ball around zero separately from the remaining domain as in e.g. Equations (6.59).
6. Compute the integrals and solve for the parameter of interest to obtain the EM update as in e.g. Equation (6.65).

For this recipe to work for other channel parameters than τ, the partial derivative must conform to a certain structure. Specifically, consider the channel defined by $p(\alpha_j; \boldsymbol{\theta}_I) = (1 - w_j\tau)\delta_{\mathrm{Dirac}}(\alpha_J) + w_j\tau f_1(\boldsymbol{\theta}_I)\exp(f_2(\alpha_j, \boldsymbol{\theta}_I))$ for some well-behaved functions f_1 and f_2. This channel has partial derivatives for each of the k parameters in $\boldsymbol{\theta}_I$

$$\frac{\partial}{\partial\theta_k}\ln(p(\alpha_j; \boldsymbol{\theta}_I)) = \frac{\partial}{\partial\theta_k}\ln((1 - w_j\tau)\delta_{\mathrm{Dirac}}(\alpha_J) + w_j\tau f_1(\boldsymbol{\theta}_I)\exp(f_2(\alpha_j, \boldsymbol{\theta}_I))) \tag{6.68}$$

$$= \frac{\frac{\partial}{\partial\theta_k} f_1(\boldsymbol{\theta}_I)\exp(f_2(\alpha_j, \boldsymbol{\theta}_I))}{(1 - w_j\tau)\delta_{\mathrm{Dirac}}(\alpha_J) + w_j\tau f_1(\boldsymbol{\theta}_I)\exp(f_2(\alpha_j, \boldsymbol{\theta}_I))} \tag{6.69}$$

$$= \begin{cases} 0, & \alpha_j = 0 \\ \frac{\frac{\partial}{\partial\theta_k} f_1(\boldsymbol{\theta}_I)\exp(f_2(\alpha_j, \boldsymbol{\theta}_I))}{w_j\tau f_1(\boldsymbol{\theta}_I)\exp(f_2(\alpha_j, \boldsymbol{\theta}_I))}, & \alpha_j \neq 0 \end{cases} \tag{6.70}$$

Now consider the case in which f_1 and f_2 have structure such that

$$0 = \sum_{j=1}^{n} \int_{\alpha_j \in \mathcal{B}_\epsilon} p(\alpha_j | \mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t) 0 d\alpha_j +$$

$$\int_{\alpha_j \in \overline{\mathcal{B}}_\epsilon} p(\alpha_j | \mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t) \frac{\frac{\partial}{\partial\theta_k} f_1(\boldsymbol{\theta}_I)\exp(f_2(\alpha_j, \boldsymbol{\theta}_I))}{w_j\tau f_1(\boldsymbol{\theta}_I)\exp(f_2(\alpha_j, \boldsymbol{\theta}_I))} d\alpha_j \tag{6.71}$$

$$= \sum_{j=1}^{n} \int_{\alpha_j \in \mathcal{B}_\epsilon} p(\alpha_j | \mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t) 0 d\alpha_j + \int_{\alpha_j \in \overline{\mathcal{B}}_\epsilon} p(\alpha_j | \mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t)(f_3(\alpha_j) - \theta_k) d\alpha_j \tag{6.72}$$

for some function f_3 which does not depend on θ_k. In this case we find that

$$0 = \sum_{j=1}^{n} \int_{\alpha_j \in \overline{\mathcal{B}}_\epsilon} p(\alpha_j | \mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t)(f_3(\alpha_j) - \theta_k) d\alpha_j \tag{6.73}$$

$$= \sum_{j=1}^{n} \pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t) \int_{\alpha_j \in \overline{\mathcal{B}}_\epsilon} \varphi_{\alpha_j | \mathbf{y}; s_j, r_j, \boldsymbol{\theta}_I^t}(\alpha_j; \boldsymbol{\theta}_I^t)(f_3(\alpha_j) - \theta_k) d\alpha_j \tag{6.74}$$

$$\theta_k^{t+1} = \frac{\sum_{j=1}^{n} \pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t) \int_{\alpha_j \in \overline{\mathcal{B}}_\epsilon} \varphi_{\alpha_j | \mathbf{y}; s_j, r_j, \boldsymbol{\theta}_I^t}(\alpha_j; \boldsymbol{\theta}_I^t) f_3(\alpha_j) d\alpha_j}{\sum_{j=1}^{n} \pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t) \int_{\alpha_j \in \overline{\mathcal{B}}_\epsilon} \varphi_{\alpha_j | \mathbf{y}; s_j, r_j, \boldsymbol{\theta}_I^t}(\alpha_j; \boldsymbol{\theta}_I^t) d\alpha_j} \tag{6.75}$$

$$= \frac{\sum_{j=1}^{n} \pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t) \int_{\alpha_j} \varphi_{\alpha_j | \mathbf{y}; s_j, r_j, \boldsymbol{\theta}_I^t}(\alpha_j; \boldsymbol{\theta}_I^t) f_3(\alpha_j) d\alpha_j}{\sum_{j=1}^{n} \pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t)} \tag{6.76}$$

$$= \frac{\sum_{j=1}^{n} \pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t) \int_{\alpha_j} \varphi_{\alpha_j | \mathbf{y}; s_j, r_j, \boldsymbol{\theta}_I^t}(\alpha_j; \boldsymbol{\theta}_I^t) f_3(\alpha_j) d\alpha_j}{\tau^{t+1} \sum_{j=1}^{n} w_j} \tag{6.77}$$

where we have assumed that all the integrands are well-behaved at zero and that τ is the first parameter to be updated. From Equation (6.77), we find that this EM update separates into a slab-part only element ($\int_{\alpha_j} \varphi_{\alpha_j | \mathbf{y}; s_j, r_j, \boldsymbol{\theta}_I^t}(\alpha_j; \boldsymbol{\theta}_I^t) f_3(\alpha_j) d\alpha_j$) and the posterior support probabilities $\pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t)$. Furthermore, since it turns out that these slab-part elements are typically computed in the GAMP channel update, all elements needed in the EM-update are already available following a GAMP channel update which allows for efficient implementations of the EM update. This is the case for the sparse Bernoulli-Gauss channel as shown in [28] and detailed in Section 6.2.2.5. The above imposed structure on the partial derivatives is somewhat limiting in terms of the possible $\varphi(\alpha_j; [\boldsymbol{\theta}_I]_j)$ that one may consider. However, as is done in some updates in [28] as well as for the sparse Bernoulli-Laplace input channel detailed in Section 6.2.2.5, one may apply various approximations to impose this structure.

An implementation in which the GWS τ EM update is decoupled from the slab-part EM updates is slightly more tricky than the corresponding channel updates since we assume common channel parameters shared across all n coefficients effectively requiring a reduction from all n elements to a single element. However, if the slab-part channel has the structure discussed above, one may store the relevant channel update elements and reuse those in the common EM updates which still provides an efficient implementation in which the GWS framework may be easily combined with different slab-part priors.

6.2.2.4 I.i.d. Sparse Bernoulli-Laplace input channel

For the i.i.d. BL input channel given in Equations (3.127) and (3.129), we consider the following parameter update order: τ, λ, μ. For the EM updates of the signal density parameter τ, we may (via Equation (3.41)) use the general result in Equation (6.51)

$$\tau^{t+1} = \frac{1}{n} \sum_{j=1}^{n} \pi(r_j, s_j, \boldsymbol{\theta}_I^t) \tag{6.78}$$

$$= \frac{1}{n} \sum_{j=1}^{n} \frac{\tau^t(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I)}{(1 - \tau^t)\mathcal{N}(0; r_j, s_j) + \tau^t(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I)} \tag{6.79}$$

for $\underline{\mathcal{Z}}_I, \bar{\mathcal{Z}}_I$ given in Equations (3.93), (3.94), respectively, and with appropriately indexed r_j, s_j in $\underline{\mathcal{Z}}_I, \bar{\mathcal{Z}}_I$.

Using some of the ideas from [40] (which address a AWLN output channel) and [41] (which address an elastic net input channel), we may find the EM updates for the remaining parameters,

λ, μ. For the EM update of λ, we have

$$\lambda^{t+1} = \underset{\lambda>0}{\arg\max} \sum_{j=1}^{n} \mathbb{E}_{\boldsymbol{\alpha}|\mathbf{y},\mathbf{s},\mathbf{r},\boldsymbol{\theta}_I^t}[\ln\Big(p(\alpha_j;\lambda,\boldsymbol{\theta}_{I\setminus\lambda}^t)\Big)] \tag{6.80}$$

$$= \underset{\lambda>0}{\arg\max} \sum_{j=1}^{n} \int_{\alpha_j} p(\alpha_j|\mathbf{y},s_j,r_j,\boldsymbol{\theta}_I^t)\ln\Bigg((1-\tau^{t+1})\delta_{\text{Dirac}}(\alpha_j) + \tau^{t+1}\frac{\lambda}{2}\exp\big(-\lambda|\alpha_j-\mu^t|\big)\Bigg) d\alpha_j \tag{6.81}$$

Setting the derivative of the objective equal to zero, we obtain

$$0 = \frac{\partial}{\partial\lambda}\sum_{j=1}^{n}\int_{\alpha_j} p(\alpha_j|\mathbf{y},s_j,r_j,\boldsymbol{\theta}_I^t)\ln\Bigg((1-\tau^{t+1})\delta_{\text{Dirac}}(\alpha_j)+\tau^{t+1}\frac{\lambda}{2}\exp\big(-\lambda|\alpha_j-\mu^t|\big)\Bigg)d\alpha_j \tag{6.82}$$

$$= \sum_{j=1}^{n}\int_{\alpha_j} p(\alpha_j|\mathbf{y},s_j,r_j,\boldsymbol{\theta}_I^t)\frac{\partial}{\partial\lambda}\ln\Bigg((1-\tau^{t+1})\delta_{\text{Dirac}}(\alpha_j)+\tau^{t+1}\frac{\lambda}{2}\exp\big(-\lambda|\alpha_j-\mu^t|\big)\Bigg)d\alpha_j \tag{6.83}$$

Where we have used Leibniz's integral rule to exchange differentiation and integration using the same argument as for Equation (6.39). Now for the partial derivative, we have

$$\frac{\partial}{\partial\lambda}\ln\Bigg((1-\tau^{t+1})\delta_{\text{Dirac}}(\alpha_j)+\tau^{t+1}\frac{\lambda}{2}\exp\big(-\lambda|\alpha_j-\mu^t|\big)\Bigg) \tag{6.84}$$

$$= \frac{\tau^{t+1}\frac{1}{2}\exp(-\lambda|\alpha_j-\mu^t|) - \tau^{t+1}\frac{\lambda}{2}\exp(-\lambda|\alpha_j-\mu^t|)\,|\alpha_j-\mu^t|}{(1-\tau^{t+1})\delta_{\text{Dirac}}(\alpha_j)+\tau^{t+1}\frac{\lambda}{2}\exp(-\lambda|\alpha_j-\mu^t|)} \tag{6.85}$$

$$= \frac{\frac{\tau^{t+1}}{2}\exp(-\lambda|\alpha_j-\mu^t|)\,(1-\lambda|\alpha_j-\mu|)}{(1-\tau^{t+1})\delta_{\text{Dirac}}(\alpha_j)+\lambda\frac{\tau^{t+1}}{2}\exp(-\lambda|\alpha_j-\mu^t|)} \tag{6.86}$$

$$= \frac{\frac{\frac{\tau^{t+1}}{2}\exp\big(-\lambda|\alpha_j-\mu^t|\big)(1-\lambda|\alpha_j-\mu|)}{(1-\tau^{t+1})\delta_{\text{Dirac}}(\alpha_j)}}{1+\frac{\lambda\frac{\tau^{t+1}}{2}\exp\big(-\lambda|\alpha_j-\mu^t|\big)}{(1-\tau^{t+1})\delta_{\text{Dirac}}(\alpha_j)}} \tag{6.87}$$

$$= \begin{cases} 0, & \alpha_j = 0 \\ \frac{1}{\lambda} - |\alpha_j-\mu^t|, & \alpha_j \neq 0 \end{cases} \tag{6.88}$$

Following [28], we may define a closed ball $\mathcal{B}_\epsilon := [-\epsilon,\epsilon]$ and its complement $\overline{\mathcal{B}}_\epsilon = \mathbb{R}\setminus\mathcal{B}_\epsilon$ which may be used to evaluate the integral in Equation (6.83) as $\epsilon \to 0$

$$0 = \sum_{j=1}^{n}\int_{\alpha_j} p(\alpha_j|\mathbf{y},s_j,r_j,\boldsymbol{\theta}_I^t)\frac{\partial}{\partial\lambda}\ln\Bigg((1-\tau^{t+1})\delta_{\text{Dirac}}(\alpha_j)+\tau^{t+1}\frac{\lambda}{2}\exp\big(-\lambda|\alpha_j-\mu^t|\big)\Bigg)d\alpha_j \tag{6.89}$$

$$= \sum_{j=1}^{n}\Bigg(\int_{\alpha_j\in\mathcal{B}_\epsilon} p(\alpha_j|\mathbf{y},s_j,r_j,\boldsymbol{\theta}_I^t)0d\alpha_j + \int_{\alpha_j\in\overline{\mathcal{B}}_\epsilon} p(\alpha_j|\mathbf{y},s_j,r_j,\boldsymbol{\theta}_I^t)(\frac{1}{\lambda}-|\alpha_j-\mu^t|)d\alpha_j\Bigg) \tag{6.90}$$

$$= \sum_{j=1}^{n}\int_{\alpha_j\in\overline{\mathcal{B}}_\epsilon} p(\alpha_j|\mathbf{y},s_j,r_j,\boldsymbol{\theta}_I^t)(\frac{1}{\lambda}-|\alpha_j-\mu^t|)d\alpha_j \tag{6.91}$$

$$= \sum_{j=1}^{n}\Bigg(\frac{1}{\lambda}\int_{\alpha_j\in\overline{\mathcal{B}}_\epsilon} p(\alpha_j|\mathbf{y},s_j,r_j,\boldsymbol{\theta}_I^t)d\alpha_j - \int_{\alpha_j\in\overline{\mathcal{B}}_\epsilon} p(\alpha_j|\mathbf{y},s_j,r_j,\boldsymbol{\theta}_I^t)|\alpha_j-\mu^t|d\alpha_j\Bigg) \tag{6.92}$$

$$\tag{6.93}$$

$$= \sum_{j=1}^{n} \left(\frac{1}{\lambda} \pi(r_j, s_j, \boldsymbol{\theta}_I^t) - \int_{\alpha_j \in \overline{\mathcal{B}}_\epsilon} p(\alpha_j | \mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t) |\alpha_j - \mu^t| d\alpha_j \right) \tag{6.94}$$

$$= \frac{1}{\lambda} n \tau^{t+1} - \sum_{j=1}^{n} \int_{\alpha_j \in \overline{\mathcal{B}}_\epsilon} p(\alpha_j | \mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t) |\alpha_j - \mu^t| d\alpha_j \tag{6.95}$$

Solving for λ, we get

$$\lambda^{t+1} = \frac{n \tau^{t+1}}{\sum_{j=1}^{n} \int_{\overline{\mathcal{B}}_\epsilon} p(\alpha_j | \mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t) |\alpha_j - \mu^t| d\alpha_j} \tag{6.96}$$

The integral in the denominator in Equation (6.96) is exactly the expectation from Equation (3.127) with four exceptions:

1. The Dirac delta contribution should be left out since the integration is over $\overline{\mathcal{B}}_\epsilon$ instead of the entire real line. This is, however, not important since the Dirac delta contribution turns out to be zero anyway.
2. The integral involves the expectation of $|\alpha_j - \mu^t|$ instead of α_j. Thus, applying a shift $\check{\alpha}_j = \alpha_j - \mu^t$ eliminates the contribution from a non-zero μ^t.
3. The absolute value in $|\alpha_j - \mu^t|$ must be addressed. Specifically, after having done the shift $\check{\alpha}_j = \alpha_j - \mu^t$, one must handle the absolute value $|\check{\alpha}_j|$ in (what corresponds to) Equation (3.102) correctly by changing the sign of the integral over the negative part of the real line.
4. The integration in (what corresponds to) (3.102) is done from $-\infty$ to 0^- and 0^+ to ∞. However, since the GAMP posterior $p(\alpha_j | \mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t)$ without the Dirac delta contribution at $\alpha_j = 0$ is well behaved, this makes no difference and we may integrate over the entire real line.

Thus, we have

$$\int_{\alpha_j \in \overline{\mathcal{B}}_\epsilon} p(\alpha_j | \mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t) |\alpha_j - \mu^t| d\alpha_j$$
$$= \tau^t \left(\frac{\bar{\mathcal{Z}}_I}{\mathcal{Z}_I} \left(\bar{r}_j + \sqrt{s_j} \frac{\phi_{\mathcal{N}}\left(\frac{\bar{r}}{\sqrt{s_j}}\right)}{\Phi_{\mathcal{N}}\left(\frac{\bar{r}_j}{\sqrt{s_j}}\right)} \right) - \frac{\underline{\mathcal{Z}}_I}{\mathcal{Z}_I} \left(\underline{r}_j - \sqrt{s_j} \frac{\phi_{\mathcal{N}}\left(\frac{-\underline{r}_j}{\sqrt{s_j}}\right)}{\Phi_{\mathcal{N}}\left(\frac{-\underline{r}_j}{\sqrt{s_j}}\right)} \right) \right) \tag{6.97}$$

for

$$\check{r}_j = r_j - \mu^t \tag{6.98}$$
$$\bar{r}_j = \check{r}_j - \lambda^t s \tag{6.99}$$
$$\underline{r}_j = \check{r}_j + \lambda^t s_j \tag{6.100}$$

and $\mathcal{Z}_I$, $\underline{\mathcal{Z}}_I$, and $\bar{\mathcal{Z}}_I$ as defined in Equations (3.92), (3.93), (3.94) (with appropriate indices on the variables), respectively. Finally, we have the EM update for λ

$$\lambda^{t+1} = \frac{n \tau^{t+1}}{\sum_{j=1}^{n} \tau^t \left(\frac{\bar{\mathcal{Z}}_I}{\mathcal{Z}_I} \left(\bar{r}_j + \sqrt{s_j} \frac{\phi_{\mathcal{N}}\left(\frac{\bar{r}}{\sqrt{s_j}}\right)}{\Phi_{\mathcal{N}}\left(\frac{\bar{r}_j}{\sqrt{s_j}}\right)} \right) - \frac{\underline{\mathcal{Z}}_I}{\mathcal{Z}_I} \left(\underline{r}_j - \sqrt{s_j} \frac{\phi_{\mathcal{N}}\left(\frac{-\underline{r}_j}{\sqrt{s_j}}\right)}{\Phi_{\mathcal{N}}\left(\frac{-\underline{r}_j}{\sqrt{s_j}}\right)} \right) \right)} \tag{6.101}$$

For the EM update of μ, we have

$$\mu^{t+1} = \arg\max_{\mu} \sum_{j=1}^{n} \mathbb{E}_{\boldsymbol{\alpha} | \mathbf{y}, \mathbf{s}, \mathbf{r}, \boldsymbol{\theta}_I^t} [\ln \left(p(\alpha_j; \mu, \boldsymbol{\theta}_{I \setminus \mu}^t) \right)] \tag{6.102}$$

$$= \arg\max_{\mu} \sum_{j=1}^{n} \int_{\alpha_j} p(\alpha_j | \mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t) \ln \Bigg($$
$$(1 - \tau^{t+1}) \delta_{\text{Dirac}}(\alpha_j) + \tau^{t+1} \frac{\lambda^{t+1}}{2} \exp(-\lambda^{t+1} |\alpha_j - \mu|) \Bigg) d\alpha_j \tag{6.103}$$

The partial derivative with respect to μ of the integrand in Equation (6.103) is unfortunately not continuous (even if using the same argument for the Delta function as in Equation (6.39)) due to the absolute value causing problems at $\alpha_j = \mu$. Thus, we may not apply Leibniz's integral rule. However, in order to derive a reasonable EM update for μ, we apply the quadratic approximation $|\alpha_j - \mu| \approx (\alpha_j - \mu)^2$. Now, using Leibniz's integral rule, taking the partial derivative of the objective, and setting equal to zero, we obtain

$$\begin{aligned} 0 &= \frac{\partial}{\partial\mu} \sum_{j=1}^{n} \int_{\alpha_j} p(\alpha_j | \mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t) \ln \Bigg((1-\tau^{t+1}) \delta_{\text{Dirac}}(\alpha_j) \\ &\quad + \tau^{t+1} \frac{\lambda^{t+1}}{2} \exp\big(-\lambda^{t+1} (\alpha_j - \mu)^2\big) \Bigg) d\alpha_j && (6.104) \\ &= \sum_{j=1}^{n} \int_{\alpha_j} p(\alpha_j | \mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t) \frac{\partial}{\partial\mu} \ln \Bigg((1-\tau^{t+1}) \delta_{\text{Dirac}}(\alpha_j) \\ &\quad + \tau^{t+1} \frac{\lambda^{t+1}}{2} \exp\big(-\lambda^{t+1} (\alpha_j - \mu)^2\big) \Bigg) d\alpha_j && (6.105) \end{aligned}$$

For the partial derivative, we have

$$\begin{aligned} &\frac{\partial}{\partial\mu} \ln \Bigg((1-\tau^{t+1}) \delta_{\text{Dirac}}(\alpha_j) + \tau^{t+1} \frac{\lambda^{t+1}}{2} \exp\big(-\lambda^{t+1} (\alpha_j - \mu)^2\big) \Bigg) d\alpha_j && (6.106) \\ &= \frac{\tau^{t+1} \frac{\lambda^{t+1}}{2} \exp\big(-\lambda^{t+1} (\alpha_j - \mu)^2\big) \left(-2\lambda^{t+1} (\alpha - \mu)(-1)\right)}{(1-\tau^{t+1}) \delta_{\text{Dirac}}(\alpha_j) + \tau^{t+1} \frac{\lambda^{t+1}}{2} \exp(-\lambda^{t+1} (\alpha_j - \mu)^2)} && (6.107) \\ &= \frac{\tau^{t+1} \frac{\lambda^{t+1}}{2} \exp\big(-\lambda^{t+1} (\alpha_j - \mu)^2\big) \left(2\lambda^{t+1} (\alpha - \mu)\right)}{(1-\tau^{t+1}) \delta_{\text{Dirac}}(\alpha_j) + \tau^{t+1} \frac{\lambda^{t+1}}{2} \exp(-\lambda^{t+1} (\alpha_j - \mu)^2)} && (6.108) \\ &= \frac{\frac{\tau^{t+1} \frac{\lambda^{t+1}}{2} \exp\big(-\lambda^{t+1} (\alpha_j - \mu)^2\big) (2\lambda^{t+1} (\alpha - \mu))}{(1-\tau^{t+1}) \delta_{\text{Dirac}}(\alpha_j)}}{1 + \frac{\tau^{t+1} \frac{\lambda^{t+1}}{2} \exp(-\lambda^{t+1} (\alpha_j - \mu)^2)}{(1-\tau^{t+1}) \delta_{\text{Dirac}}(\alpha_j)}} && (6.109) \\ &= \begin{cases} 0, & \alpha_j = 0 \\ 2\lambda^{t+1} (\alpha_j - \mu), & \alpha_j \neq 0 \end{cases} && (6.110) \end{aligned}$$

Again, using the closed ball $\mathcal{B}_\epsilon$, we may evaluate the integral in Equation (6.105) as $\epsilon \to 0$

$$\begin{aligned} 0 &= \sum_{j=1}^{n} \int_{\alpha_j} p(\alpha_j | \mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t) \frac{\partial}{\partial\mu} \ln \Bigg((1-\tau^{t+1}) \delta_{\text{Dirac}}(\alpha_j) \\ &\quad + \tau^{t+1} \frac{\lambda^{t+1}}{2} \exp\big(-\lambda^{t+1} (\alpha_j - \mu)^2\big) \Bigg) d\alpha_j && (6.111) \\ &= \sum_{j=1}^{n} \Bigg(\int_{\alpha_j \in \mathcal{B}_\epsilon} p(\alpha_j | \mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t) 0 d\alpha_j && (6.112) \\ &\quad + \int_{\alpha_j \in \overline{\mathcal{B}}_\epsilon} p(\alpha_j | \mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t) (2\lambda^{t+1} (\alpha_j - \mu)) d\alpha_j \Bigg) && (6.113) \\ &= \sum_{j=1}^{n} \int_{\alpha_j \in \overline{\mathcal{B}}_\epsilon} p(\alpha_j | \mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t) (\alpha_j - \mu) d\alpha_j && (6.114) \\ &= \sum_{j=1}^{n} \int_{\alpha_j \in \overline{\mathcal{B}}_\epsilon} p(\alpha_j | \mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t) \alpha_j d\alpha_j - \mu \sum_{j=1}^{n} \int_{\alpha_j \in \overline{\mathcal{B}}_\epsilon} p(\alpha_j | \mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t) d\alpha_j && (6.115) \\ &= \sum_{j=1}^{n} \int_{\alpha_j \in \overline{\mathcal{B}}_\epsilon} p(\alpha_j | \mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t) \alpha_j d\alpha_j - \mu \pi(r_j, s_j, \boldsymbol{\theta}_I^t) && (6.116) \end{aligned}$$

Thus, we have the following update of μ

$$\mu^{t+1} = \frac{\sum_{j=1}^{n} \int_{\alpha_j \in \overline{\mathcal{B}}_\epsilon} p(\alpha_j|\mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t)\alpha_j d\alpha_j}{\pi(r_j, s_j, \boldsymbol{\theta}_I^t)} \tag{6.117}$$

$$= \frac{\sum_{j=1}^{n} \int_{\alpha_j \in \overline{\mathcal{B}}_\epsilon} p(\alpha_j|\mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t)\alpha_j d\alpha_j}{n\tau^{t+1}} \tag{6.118}$$

The numerator in Equation (6.118) is the expectation from Equation (3.127) with the exception of leaving out $\alpha_j = 0$ in the integration. However, since the remaining part of the GAMP posterior $p(\alpha_j|\mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t)$ (leaving out the Dirac delta contribution) is well-behaved, we find that

$$\int_{\alpha_j \in \overline{\mathcal{B}}_\epsilon} p(\alpha_j|\mathbf{y}, s_j, r_j, \boldsymbol{\theta}_I^t)\alpha_j d\alpha_j \tag{6.119}$$

$$= \tau^t \left(\mu^t \frac{(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I)}{\mathcal{Z}_I} + \frac{\underline{\mathcal{Z}}_I}{\mathcal{Z}_I} \left(\underline{r}_j - \sqrt{s_j} \frac{\phi_{\mathcal{N}}\left(\frac{-\underline{r}_j}{\sqrt{s_j}}\right)}{\Phi_{\mathcal{N}}\left(\frac{-\underline{r}_j}{\sqrt{s_j}}\right)} \right) + \frac{\bar{\mathcal{Z}}_I}{\mathcal{Z}_I} \left(\bar{r}_j + \sqrt{s_j} \frac{\phi_{\mathcal{N}}\left(\frac{\bar{r}_j}{\sqrt{s_j}}\right)}{\Phi_{\mathcal{N}}\left(\frac{\bar{r}_j}{\sqrt{s_j}}\right)} \right) \right) \tag{6.120}$$

for

$$\check{r}_j = r_j - \mu^t \tag{6.121}$$

$$\bar{r}_j = \check{r}_j - \lambda^t s \tag{6.122}$$

$$\underline{r}_j = \check{r}_j + \lambda^t s_j \tag{6.123}$$

and $\mathcal{Z}_I$, $\underline{\mathcal{Z}}_I$, and $\bar{\mathcal{Z}}_I$ as defined in Equations (3.92), (3.93), (3.94) (with appropriate indices on the variables), respectively. Finally, we have the EM update for μ

$$\mu^{t+1} = \frac{\sum_{j=1}^{n} \tau^t \left(\mu^t \frac{(\underline{\mathcal{Z}}_I + \bar{\mathcal{Z}}_I)}{\mathcal{Z}_I} + \frac{\underline{\mathcal{Z}}_I}{\mathcal{Z}_I} \left(\underline{r}_j - \sqrt{s_j} \frac{\phi_{\mathcal{N}}\left(\frac{-\underline{r}_j}{\sqrt{s_j}}\right)}{\Phi_{\mathcal{N}}\left(\frac{-\underline{r}_j}{\sqrt{s_j}}\right)} \right) + \frac{\bar{\mathcal{Z}}_I}{\mathcal{Z}_I} \left(\bar{r}_j + \sqrt{s_j} \frac{\phi_{\mathcal{N}}\left(\frac{\bar{r}_j}{\sqrt{s_j}}\right)}{\Phi_{\mathcal{N}}\left(\frac{\bar{r}_j}{\sqrt{s_j}}\right)} \right) \right)}{n\tau^{t+1}} \tag{6.124}$$

Based on the results in Equations (6.101) and (6.124), we may identify the following Laplace EM updates to be used in the GWS EM update framework described in Section 6.2.2.2

$$\lambda^{t+1} = \frac{\tau^{t+1} \sum_{j=1}^{n} w_j}{\sum_{j=1}^{n} \pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t) \left(\frac{\bar{\mathcal{Z}}_I}{\mathcal{Z}_{\varphi_j}} \left(\bar{r}_j + \sqrt{s_j} \frac{\phi_{\mathcal{N}}\left(\frac{\bar{r}}{\sqrt{s_j}}\right)}{\Phi_{\mathcal{N}}\left(\frac{\bar{r}_j}{\sqrt{s_j}}\right)} \right) - \frac{\underline{\mathcal{Z}}_I}{\mathcal{Z}_{\varphi_j}} \left(\underline{r}_j - \sqrt{s_j} \frac{\phi_{\mathcal{N}}\left(\frac{-\underline{r}_j}{\sqrt{s_j}}\right)}{\Phi_{\mathcal{N}}\left(\frac{-\underline{r}_j}{\sqrt{s_j}}\right)} \right) \right)} \tag{6.125}$$

$$\mu^{t+1} = \frac{\sum_{j=1}^{n} \pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t) \left(\mu^t + \frac{\underline{\mathcal{Z}}_I}{\mathcal{Z}_{\varphi_j}} \left(\underline{r}_j - \sqrt{s_j} \frac{\phi_{\mathcal{N}}\left(\frac{-\underline{r}_j}{\sqrt{s_j}}\right)}{\Phi_{\mathcal{N}}\left(\frac{-\underline{r}_j}{\sqrt{s_j}}\right)} \right) + \frac{\bar{\mathcal{Z}}_I}{\mathcal{Z}_{\varphi_j}} \left(\bar{r}_j + \sqrt{s_j} \frac{\phi_{\mathcal{N}}\left(\frac{\bar{r}_j}{\sqrt{s_j}}\right)}{\Phi_{\mathcal{N}}\left(\frac{\bar{r}_j}{\sqrt{s_j}}\right)} \right) \right)}{\tau^{t+1} \sum_{j=1}^{n} w_j} \tag{6.126}$$

6.2.2.5 I.i.d. Sparse Bernoulli-Gauss Input Channel

For the i.i.d. BG input channel given in Equations (3.154) and (3.155), Krzakala et al. suggested the following EM recursions on the channel parameters [5] (Eqs. (74), (78), (79))[6]

$$\tau^{t+1} = \frac{\sum_{j=1}^{n} \frac{\frac{1}{\tilde{\theta}^t}+\frac{1}{s_j^{t+1}}}{\frac{r_j^{t+1}}{s_j^{t+1}}+\frac{\bar{\theta}^t}{\tilde{\theta}^t}} \bar{\alpha}_j^{t+1}}{\sum_{j=1}^{n} \left[1-\tau^t+\frac{\tau^t}{\sqrt{\tilde{\theta}^t}}\sqrt{\frac{1}{\tilde{\theta}^t}+\frac{1}{s_j^{t+1}}}\exp\left(\frac{\left(\frac{r_j^{t+1}}{s_j^{t+1}}+\frac{\bar{\theta}^t}{\tilde{\theta}^t}\right)^2}{2\left(\frac{1}{\tilde{\theta}^t}+\frac{1}{s_j^{t+1}}\right)}-\frac{(\bar{\theta}^t)^2}{2\tilde{\theta}^t}\right)\right]^{-1}} \tag{6.127}$$

$$\bar{\theta}^{t+1} = \frac{\sum_{j=1}^{n} \bar{\alpha}_j^{t+1}}{n\tau^{t+1}} \tag{6.128}$$

$$\tilde{\theta}^{t+1} = \frac{\sum_{j=1}^{n} (\tilde{\alpha}_j^{t+1}+(\bar{\alpha}_j^{t+1})^2)}{n\tau^{t+1}} - (\bar{\theta}^{t+1})^2 \tag{6.129}$$

Note that Krzakala et al. do not make it clear (in either of [5], [6]) whether or not the iteration dependence on the channel parameters (and thereby the ordering of the updates) are as given in Equations (6.127) - (6.129). In [5] it is noted that the following heuristic rules should be used together with the Equations (6.127), (6.128), (6.129)

- If the variance, $\tilde{\theta}^{t+1}$, becomes negative, it should be set to zero.
- If the signal density, τ^{t+1}, becomes larger than the undersampling ratio δ, it should be set to δ.
- A damping of 0.5 should be used on all EM updates. That is, the updated parameter values should be taken to be the mean of the values of the updates in Equations (6.127), (6.128), (6.129) and the respective previous values.

Schniter and Vila suggested the following EM recursions on the channel parameters for the i.i.d. BG input channel [28] (Eqs. (34), (41), (47)), [39] (Eqs. (19), (25), (32))

$$\tau^{t+1} = \frac{1}{n}\sum_{j} \mathscr{f}(r_j^{t+1}, s_j^{t+1}; \tau^t, \bar{\theta}^t, \tilde{\theta}^t) \tag{6.130}$$

$$\bar{\theta}^{t+1} = \frac{1}{n\tau^{t+1}}\sum_{j=1}^{n} \mathscr{f}(r_j^{t+1}, s_j^{t+1}; \tau^{t+1}, \bar{\theta}^t, \tilde{\theta}^t)\mathscr{g}(r_j^{t+1}, s_j^{t+1}; \tau^{t+1}, \bar{\theta}^t, \tilde{\theta}^t) \tag{6.131}$$

$$\begin{aligned}\tilde{\theta}^{t+1} = \frac{1}{n\tau^{t+1}}\sum_{j=1}^{n} \mathscr{f}(r_j^{t+1}, s_j^{t+1}; \tau^{t+1}, \bar{\theta}^{t+1}, \tilde{\theta}^t)\Big(& \\ |\bar{\theta}^{t+1} - \mathscr{g}(r_j^{t+1}, s_j^{t+1}; \tau^{t+1}, \bar{\theta}^{t+1}, \tilde{\theta}^t)|^2 + \mathscr{h}(r_j^{t+1}, s_j^{t+1}; \tau^{t+1}, \bar{\theta}^{t+1}, \tilde{\theta}^t)\Big)&\end{aligned} \tag{6.132}$$

Where $\mathscr{f}, \mathscr{g}, \mathscr{h}$ are as defined in Equations (3.171), (3.166), and (3.167), respectively.

Based on the results in Equations (6.131) and (6.132), we may identify the following Gauss EM

[6] Eq. (74) in [6] lacks a set of parentheses to be equal to Equation (6.127). However, Eqs. (71) and (72) in [6] suggest that Eq. (74) is to be interpreted as in Equation (6.127).

updates to be used in the GWS EM update framework described in Section 6.2.2.2

$$\bar{\theta}^{t+1} = \frac{\sum_{j=1}^{n} \left(\pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t) \frac{\frac{\bar{\theta}_j^t}{\tilde{\theta}_j^t} + \frac{r_j}{s_j}}{\frac{1}{\tilde{\theta}_j^t} + \frac{1}{s_j}} \right)}{\tau^{t+1} \sum_{j=1}^{n} w_j} \tag{6.133}$$

$$\tilde{\theta}^{t+1} = \frac{\sum_{j=1}^{n} \pi_j^{\mathrm{w}}(r_j, s_j, \boldsymbol{\theta}_I^t) \left(\left| \bar{\theta}^{t+1} - \left(\frac{\frac{\bar{\theta}_j^{t+1}}{\tilde{\theta}_j^t} + \frac{r_j}{s_j}}{\frac{1}{\tilde{\theta}_j^t} + \frac{1}{s_j}} \right) \right|^2 + \frac{1}{\tilde{\theta}_j^t + \frac{1}{s_j}} \right)}{\tau^{t+1} \sum_{j=1}^{n} w_j} \tag{6.134}$$

$$\tag{6.135}$$

6.3 Parameter Initialisation

Since the GAMP and EM algorithms are only guaranteed to converge to local optima, proper parameter initialisation is important. The various proposed ways to initialise the GAMP states are reproduced in the respective GAMP implementations in Algorithms 1, 2, and 3. Below we summarise the various proposed EM initialisations.

6.3.1 EM Initialisation of the AWGN Output Channel Parameters

For the AWGN output channel EM update in Equation (6.32), Vila and Schniter [28] (Eq. (71)) suggested the following initialisation

$$\sigma_0^2 = \frac{||\mathbf{y}||_2^2}{m(\mathrm{SNR}^0 + 1)} \tag{6.136}$$

For some assumed true signal-to-noise ratio $\mathrm{SNR}^0 = \frac{||\mathbf{A}\boldsymbol{\alpha}||_2^2}{||\mathbf{e}||_2^2}$. In lack of knowledge of the true signal-to-noise ratio, $\mathrm{SNR}^0 = 100$ is proposed.

6.3.2 EM Initialisation of the AWLN Output Channel Parameters

For the AWLN output channel EM update in Equation (6.34), Vila and Schniter [40] (Eq. (67)) suggested the following initialisation

$$\lambda_0 = 1 \tag{6.137}$$

6.3.3 EM initialisation of the i.i.d. Sparse Bernoulli-Gauss Input Channel Parameters

For the i.i.d. BG input channel EM updates in Equations (6.127) - (6.129), Krzakala et al. [5] (Eq. (80)) suggested the following initialisation

$$\tau_0 = \frac{\delta}{10} \tag{6.138}$$

$$\bar{\theta}_0 = 0 \tag{6.139}$$

$$\tilde{\theta}_0 = \frac{||\mathbf{y}||_2^2}{\tau_0 ||\mathbf{A}||_F^2} \tag{6.140}$$

Vila and Schniter [39] (Eqs. (39), (40)), [28] (Eqs. (70), (71)) suggested the following initialisation

$$\tau_0 = \delta \rho_{\mathrm{SE}}(\delta) \tag{6.141}$$

$$\bar{\theta}_0 = 0 \tag{6.142}$$

$$\tilde{\theta}_0 = \frac{||\mathbf{y}||_2^2 - m\sigma_0^2}{\tau_0 ||\mathbf{A}||_F^2} \tag{6.143}$$

when using the AWGN output channel EM initialisation in Equation (6.136) and with ρ_{SE} the theoretical LASSO phase transition curve [2] given by

$$\rho_{\text{SE}}(\delta) = \max_{c>0} \frac{1 - \frac{\zeta}{\delta}[(1+c^2)\Phi_{\mathcal{N}}(-c) - c\phi_{\mathcal{N}}(c)]}{1 + c^2 - 2[(1+c^2)\Phi_{\mathcal{N}}(-c) - c\phi_{\mathcal{N}}(c)]} \tag{6.144}$$

for $\zeta = 2$ (sparse signed vectors). When using the AWLN output channel, Schniter and Vila suggested using $\sigma_0^2 = 1$ [40] (Eq. (67)).

6.3.4 EM Initialisation of the i.i.d. Sparse Bernoulli-Laplace Input Channel Parameters

For the i.i.d. BL input channel EM updates in Equations (6.127), (6.101), and (6.124) when used with an AWGN output channel, we suggest the following initialisation

$$\tau_0 = \delta\rho_{\text{SE}}(\delta) \tag{6.145}$$

$$\mu_0 = 0 \tag{6.146}$$

$$\lambda_0 = \sqrt{\frac{2}{\frac{||\mathbf{y}||_2^2 - m\sigma_0^2}{\tau_0||\mathbf{A}||_F^2}}} \tag{6.147}$$

That is, an initialisation based on Equations (6.141)-(6.143) but with λ initialised based on the variance of a Laplace distributed random variable being $\frac{2}{\lambda^2}$.

7 GAMP Software

Various software packages include implementations of the GAMP algorithms described in this note. Here we briefly describe a few of them.

7.1 Magni GAMP Implementation

Magni is a Python package which enables reconstruction of undersampled Atomic Force Microscopy (AFM) images [58]. The `magni.cs.reconstruction.gamp` and `magni.cs.reconstruction.amp` subpackages, which are part of Magni version $\geq$ 1.6.0, provide an implementation of GAMP in Python by the authors of the present tech report. The Magni package is fully documented, has an extensive test suite, makes use of an input validation framework [59], and comes with tools for aiding in making computational results reproducible [60]. Related links are:

- Online documentation: `http://magni.readthedocs.io/en/latest/`
- Official source repository: `http://dx.doi.org/10.5278/VBN/MISC/Magni`
- GitHub repository: `https://github.com/SIP-AAU/Magni`

7.1.1 Magni GAMP Overview

The `magni.cs.reconstruction.amp` subpackage provides an implementation of the AMP algorithm by Donoho/Maleki/Monatnari described in Section 2.1. As of Magni version 1.7.0, the subpackage consists of the following modules:

- `_algorithm`: The base algorithm implementation which is available through `magni.cs.reconstruction.amp.run`.
- `_config`: Configuration module available through `magni.cs.reconstruction.amp.config` for choosing stop criterion, max iterations, threshold, etc.
- `stop_criterion`: Implementations of the stop criteria discussed in Section 5.1.
- `threshold_operator`: Implementations of threshold operators for DMM AMP as discussed in Section 2.1.
- `util`: Utilities for use in the AMP algorithm.

The `magni.cs.reconstruction.gamp` subpackage provides an implementation of the MMSE GAMP algorithm detailed in Algorithm 1 and the MMSE GAMP with Rangan sum approximations detailed in Algorithm 3. For both algorithms it also offers the damping option from [20]. As of Magni version 1.7.0, the GAMP subpackage consists of the following modules:

- `_algorithm`: The base algorithm implementation which is available through `magni.cs.reconstruction.gamp.run`.
- `_config`: Configuration module available through `magni.cs.reconstruction.gamp.config` for choosing stop criterion, max iterations, in- and output channels, etc.
- `channel_initialisation`: Implementations of the in- and output channel EM initialisations described in Section 6.3.
- `input_channel`: Implementations of the input channels discussed in Sections 3.1, 3.4, 3.5, and 3.7 with EM updates as discussed in Section 6.2.2.
- `output_channel`: Implementations of the output channels discussed in Sections 3.2 and 3.6 with EM updates as discussed in Section 6.2.1
- `stop_criterion`: Implementations of the stop criteria discussed in Section 5.1.

7.2 GAMPMatlab Implementation

The GAMPMatlab Toolbox is an implementation of GAMP in MATLAB [61]. The GAMPMatlab Toolbox is maintained by Phillip Schniter and Sundeep Rangan and has contributions from several coauthors of the various GAMP algorithms and extensions. Related links are:

- Online documentation: `http://gampmatlab.wikia.com/wiki/Generalized_Approximate_Message_Passing`
- Official source repository: `https://sourceforge.net/projects/gampmatlab/`
- SVN repository: `svn.code.sf.net/p/gampmatlab/code/`

7.3 BPCS AMP Implementation

The BPCS AMP package is another MATLAB based implementation of AMP which has been developed by Jean Barbier in connection with his PhD studies [62]. Related links are:

- GitHub repository: `https://github.com/jeanbarbier/BPCS/`

7.4 Vampyre

Vampyre is a joint collaboration on a Python implementation of GAMP algorithms by many of the authors of the works on GAMP. It has yet to kick-off, but may potentially become the reference GAMP implementation.

- GitHub repository `https://github.com/GAMPTeam/vampyre`

8 GAMP Extensions

A significant number of works on various extensions of the GAMP algorithm for specific applications have been published. Here, as a reference, we list (in no particular order) some of these works.

- Markov-tree / Markov-random-field priors [33], [63], [64].
- Learning based priors [65]
- Phase retrieval [66].
- Hyperspectral image unmixing [67], [68].
- Linearly constrained non-neagtive sparse signals [40], [69].
- Non-stationary signals [70], [71].
- Multiple measurement vectors [72], [73].
- Classification and feature selection [74], [75].
- Low rank matrix completion [76].
- Spatially coupled structured operators [77].
- Total Variation like prior [78]
- Analysis compressive sensing [79].
- Quantized measurements [80], [81].
- Magnetic resonance imaging [82].

References

[1] S. Rangan, "Generalized Approximate Message Passing for Estimation with Random Linear Mixing," in *IEEE International Symposium on Information Theory (ISIT)*, St. Petersburg, Russia, Jul. 31 – Aug. 5, 2011, pp. 2168–2172. doi:10.1109/ISIT.2011.6033942

[2] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, no. 45, p. 18914–18919, Nov. 2009. doi:10.1073/pnas.0909892106

[3] ——, "Message Passing Algorithms for Compressed Sensing: I. Motivation and Construction," in *IEEE Information Theory Workshop (ITW)*, Cairo, Egypt, Jan. 6 – 8, 2010, p. 5. doi:10.1109/ITWKSPS.2010.5503193

[4] ——, "Message Passing Algorithms for Compressed Sensing: II. Analysis and Validation," in *IEEE Information Theory Workshop (ITW)*, Cairo, Egypt, Jan. 6 – 8, 2010, p. 5. doi:10.1109/ITWKSPS.2010.5503228

[5] F. Krzakala, M. Mézard, F. Sausset, Y. Sun, and L. Zdeborová, "Probabilistic reconstruction in compressed sensing: algorithms, phase diagrams, and threshold achieving matrices," *Journal of Statistical Mechanics: Theory and Experiment*, vol. P08009, pp. 1–57, Aug. 2012. doi:10.1088/1742-5468/2012/08/P08009

[6] F. Krzakala, M. Mézard, F. Sausset, Y. F. Sun, and L. Zdeborová, "Statistical-Physics-Based Reconstruction in Compressed Sensing," *Physical Review X*, vol. 2, no. 2, pp. (021 005–1)–(021 005–18), May 2012. doi:10.1103/PhysRevX.2.021005

[7] D. Donoho and J. Tanner, "Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing," *Philosophical Transactions of the Royal Society A*, vol. 367, no. 1906, pp. 4273–4293, Nov. 2009. doi:10.1098/rsta.2009.0152

[8] L. Zdeborová and F. Krzakala, "Statistical physics of inference: Thresholds and algorithms," Jul. 2016, arxiv:1511.02476v4.

[9] A. Montanari, "Graphical models concepts in compressed sensing," in *Compressed Sensing: Theory and Applications*, Y. C. Eldar and G. Kutyniok, Eds. Cambridge University Press, 2012, ch. 9, pp. 394–438.

[10] M. Bayati and A. Montanari, "The Dynamics of Message Passing on Dense Graphs, with Applications to Compressed Sensing," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 764–785, Feb. 2011. doi:10.1109/TIT.2010.2094817

[11] M. Bayati, M. Lelarge, and A. Montanari, "Universality In Polytope Phase Transitions And Message Passing Algorithms," *The Annals of Applied Probability*, vol. 25, no. 2, pp. 753–822, Feb. 2015. doi:10.1214/14-AAP1010

[12] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor Graphs and the Sum-Product Algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001. doi:10.1109/18.910572

[13] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, 1st ed., ser. Morgan Kaufmann Series in Representation and Reasoning. San Francisco, California, USA: Morgan Kaufmann Publishers, Inc., 1988.

[14] F. Krzakala, A. Manoel, and E. W. Tramel, "Variational Free Energies for Compressed Sensing," in *IEEE International Symposium on Information Theory (ISIT)*, Honolulu, Hawaii, USA, Jun. 29 – Jul. 4, 2014, pp. 1499–1503. doi:10.1109/ISIT.2014.6875083

[15] J. T. Parker, "Approximate Message Passing Algorithms for Generalized Bilinear Inference," Ph.D. dissertation, Graduate School of The Ohio State University, 2014.

[16] S. Rangan, "Generalized Approximate Message Passing for Estimation with Random Linear Mixing," Aug. 2012, arXiv:1010.5141v2.

[17] M. Bayati, M. Lelarge, and A. Montanari, "Universality in Polytope Phase Transitions and Iterative Algorithms," in *IEEE International Symposium on Information Theory (ISIT)*, Cambridge, Massachusetts, USA, Jul. 1 – 6, 2012, pp. 1643–1647. doi:10.1109/ISIT.2012.6283554

[18] A. Javanmard and A. Montanari, "State evolution for general approximate message passing algorithms, with applications to spatial coupling," *Information and Inference*, vol. 2, no. 2, pp. 115–144, Dec. 2013. doi:10.1093/imaiai/iat004

[19] C. Rush and R. Venkataramanan, "Finite-Sample Analysis of Approximate Message Passing," in *IEEE International Symposium on Information Theory (ISIT)*, Barcelona, Spain, Jul. 10 – 15, 2016, pp. 755–759. doi:10.1109/ISIT.2016.7541400

[20] S. Rangan, P. Schniter, and A. Fletcher, "On the Convergence of Approximate Message Passing with Arbitrary Matrices," in *IEEE International Symposium on Information Theory (ISIT)*, Honolulu, Hawaii, USA, Jun. 29 – Jul. 4, 2014, pp. 236–240. doi:10.1109/ISIT.2014.6874830

[21] B. Cakmak, O. Winther, and B. H. Fleury, "S-AMP: Approximate Message Passing for General Matrix Ensembles," in *IEEE Information Theory Workshop (ITW)*, Hobart, Tasmania, Australia, Nov. 2 – 5, 2014, pp. 192–196. doi:10.1109/ITW.2014.6970819

[22] ——, "S-AMP for Non-linear Observation Models," in *IEEE International Symposium on Information Theory (ISIT)*, Hong Kong, China, Jun. 15 – 19, 2015, pp. 2807–2811. doi:10.1109/ISIT.2015.7282968

[23] S. Rangan, A. K. Fletcher, P. Schniter, and U. S. Kamilov, "Inference for Generalized Linear Models via Alternating Directions and Bethe Free Energy Minimization," *IEEE Transactions on Information Theory*, vol. 63, no. 1, pp. 676–697, Jan. 2017. doi:10.1109/TIT.2016.2619373

[24] J. Ma and L. Peng, "Orthogonal AMP," *IEEE Access*, vol. 5, pp. 2020–2033, Jan. 2017. doi:10.1109/ACCESS.2017.2653119

[25] S. Rangan, P. Schniter, E. Riegler, A. Fletcher, and V. Cevher, "Fixed Points of Generalized Approximate Message Passing with Arbitrary Matrices," in *IEEE International Symposium on Information Theory (ISIT)*, Istanbul, Turkey, Jul. 7–12, 2013, pp. 664–668. doi:10.1109/ISIT.2013.6620309

[26] H. Monajemi, S. Jafarpour, M. Gavish, S. C. . Collaboration, and D. L. Donoho, "Deterministic matrices matching the compressed sensing phase transitions of Gaussian random matrices," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 110, no. 4, p. 1181–1186, Jan. 2013. doi:10.1073/pnas.1219540110

[27] Y. Wu and S. Verdú, "Optimal Phase Transitions in Compressed Sensing," *IEEE Transactions on Information Theory*, vol. 58, no. 10, pp. 6241–6263, Oct. 2012. doi:10.1109/TIT.2012.2205894

[28] J. P. Vila and P. Schniter, "Expectation-Maximization Gaussian-Mixture Approximate Message Passing," *IEEE Transactions on Signal Processing*, vol. 61, no. 19, pp. 4658–4672, Oct. 2013. doi:10.1109/TSP.2013.2272287

[29] H. Rauhut, "Compressive Sensing and Structured Random Matrices," in *Theoretical Foundations and Numerical Methods for Sparse Recovery*, ser. Radon Series on Computational and Applied Mathematics, M. Fornasier, Ed. De Gruyter, 2010, vol. 9, pp. 1–92.

[30] T. T. Do, L. Gan, N. H. Nguyen, and T. D. Tran, "Fast and Efficient Compressive Sensing Using Structurally Random Matrices," *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 139–154, Jan. 2012. doi:10.1109/TSP.2011.2170977

[31] J. Vila, P. Schniter, S. Rangan, F. Krzakala, and L. Zdeborová, "Adaptive Damping and Mean Removal for the Generalized Approximate Message Passing Algorithm," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, South Brisbane, Queensland, Australia, Apr. 19 – 24, 2015, pp. 2021–2025. doi:10.1109/ICASSP.2015.7178325

[32] S. Som and P. Schniter, “Compressive Imaging Using Approximate Message Passing and a Markov-Tree Prior,” *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3439–3448, Jul. 2012. doi:10.1109/TSP.2012.2191780

[33] P. Schniter, “Turbo Reconstruction of Structured Sparse Signals,” in *44th Annual Conference on Information Sciences and Systems (CISS)*, Princeton, New Jersey, USA, Mar. 17 – 19, 2010, p. 6. doi:10.1109/CISS.2010.5464920

[34] A. Maleki and A. Montanari, “Analysis of Approximate Message Passing Algorithm,” in *44th Annual Conference on Information Sciences and Systems (CISS)*, Princeton, New York, USA, Mar. 17–19, 2010. doi:10.1109/CISS.2010.5464887

[35] M. R. Andersen, “Sparse inference using approximate message passing,” Master’s thesis, Technical University of Denmark, Department of Applied Mathematics and Computer Science, Matematiktorvet, Building 303B, DK-2800 Kgs. Lyngby, Denmark, 2014.

[36] I. Daubechies, M. Defrise, and C. D. Mol, “An Iterative Thresholding Algorithm for Linear Inverse Problems with a Sparsity Constraint,” *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, Nov. 2004. doi:10.1002/cpa.20042

[37] A. Beck and M. Teboulle, “A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems,” *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, Mar. 2009. doi:10.1137/080716542

[38] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers,” *Foundations and Trends(R) in Machine Learning*, vol. 3, no. 1, pp. 1–122, Jul. 2011. doi:10.1561/2200000016

[39] J. Vila and P. Schniter, “Expectation-Maximization Bernoulli-Gaussian Approximate Message Passing,” in *Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, Pacific Grove, California, USA, Nov. 6 – 9 2011, pp. 799–803. doi:10.1109/ACSSC.2011.6190117

[40] J. P. Vila and P. Schniter, “An Empirical-Bayes Approach to Recovering Linearly Constrained Non-Negative Sparse Signals,” *IEEE Transactions on Signal Processing*, vol. 62, no. 18, pp. 4689–4703, Sep. 2014. doi:10.1109/TSP.2014.2337841

[41] J. Ziniel, “Message Passing Approaches to Compressive Inference Under Structured Signal Priors,” Ph.D. dissertation, Graduate School of The Ohio State University, 2014.

[42] N. L. Johnson, S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions*, 2nd ed., ser. Wiley Series in Probability and Mathematical Statistics. Wiley-Interscience, 1994, vol. 1.

[43] D. R. Barr and E. T. Sherrill, “Mean and Variance of Truncated Normal Distributions,” *The American Statistician*, vol. 53, no. 4, pp. 357–361, Nov. 1999. doi:http://doi.org/10.2307/2686057

[44] M. R. Zaghloul, “On the calculation of the Voigt line profile: a single proper integral with a damped sine integrand,” *Monthly Notices of the Royal Astronomical Society*, vol. 375, no. 3, pp. 1043–1048, Mar. 2007. doi:10.1111/j.1365-2966.2006.11377.x

[45] J. Vila and P. Schniter, “Expectation-Maximization Gaussian-Mixture Approximate Message Passing,” in *46th Annual Conference on Information Sciences and Systems (CISS)*, Princeton, New Jersey, USA, Mar. 21 – 23, 2012, p. 6. doi:10.1109/CISS.2012.6310932

[46] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations.* SIAM, 1995.

[47] A. Maleki and D. L. Donoho, “Optimally Tuned Iterative Reconstruction Algorithms for Compressed Sensing,” *IEEE Journal Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 330–341, Apr. 2010. doi:10.1109/JSTSP.2009.2039176

[48] F. Caltagirone, L. Zdeborová, and F. Krzakala, “On Convergence of Approximate Message Passing,” in *IEEE International Symposium on Information Theory (ISIT)*, Honolulu, Hawaii, USA, Jun. 29 – Jul. 4, 2014. doi:10.1109/ISIT.2014.6875146

[49] A. Manoel, F. Krzakala, E. W. Tramel, and L. Zdeborová, “Swept Approximate Message Passing for Sparse Estimation,” in *32nd International Conference on Machine Learning (ICML)*, Lille, France, Jul. 6 – 11, 2015, p. 1123–1132.

[50] J. P. Vila, “Empirical-Bayes Approaches to Recovery of Structured Sparse Signals via Approximate Message Passing,” Ph.D. dissertation, Graduate School of The Ohio State University, 2015.

[51] U. Kamilov, S. Rangan, A. Fletcher, and M. Unser, “Approximate Message Passing with Consistent Parameter Estimation and Applications to Sparse Learning,” in *Advances in Neural Information Processing Systems (NIPS) 25*, P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds. Lake Tahoe, California, USA: MIT Press, Dec. 3 – 6, 2012, pp. 2447–2455.

[52] U. S. Kamilov, S. Rangan, A. K. Fletcher, and M. Unser, “Approximate Message Passing With Consistent Parameter Estimation and Applications to Sparse Learning,” *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2969–2985, May 2014. doi:10.1109/TIT.2014.2309005

[53] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, “Sparse Solution of Underdetermined Systems of Linear Equations by Stagewise Orthogonal Matching Pursuit,” *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1094–1121, Feb. 2012. doi:10.1109/TIT.2011.2173241

[54] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum Likelihood from Incomplete Data via the EM Algorithm,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[55] C. M. Bishop, *Pattern Recognition and Machine Learning*, ser. Information Science and Statistics. Springer, 2006.

[56] R. M. Neal and G. E. Hinton, “A View of the EM Algorithm that Justifies Incremental, Sparse, and Other Variants,” in *Learning in Graphical Models*, M. I. Jordan, Ed. MIT Press, 1999, pp. 355–368. doi:10.1007/978-94-011-5014-9_12

[57] X.-L. Meng and D. B. Rubin, “Maximum Likelihood Estimation via the ECM Algorithm: A General Framework,” *Biometrika*, vol. 80, no. 2, pp. 267–278, Jun. 1993. doi:10.2307/2337198

[58] C. S. Oxvig, P. S. Pedersen, T. Arildsen, J. Østergaard, and T. Larsen, “Magni: A Python Package for Compressive Sampling and Reconstruction of Atomic Force Microscopy Images,” *Journal of Open Research Software*, vol. 2, no. 1, p. e29, Oct. 2014. doi:10.5334/jors.bk

[59] P. S. Pedersen, C. S. Oxvig, J. Østergaard, and T. Larsen, “Validating Function Arguments in Python Signal Processing Applications,” in *Proceedings of the 15th Python in Science Conference*, Austin, Texas, USA, Jul. 11 – 17, 2016, pp. 106–113.

[60] C. S. Oxvig, T. Arildsen, and T. Larsen, “Storing Reproducible Results from Computational Experiments using Scientific Python Packages,” in *Proceedings of the 15th Python in Science Conference*, Austin, Texas, USA, Jul. 11 – 17, 2016, pp. 45–50.

[61] J. Ziniel, S. Rangan, and P. Schniter, “A Generalized Framework for Learning and Recovery of Structured Sparse Signals,” in *IEEE Statistical Signal Processing Workshop (SSP)*, Ann Arbor, Michigan, USA, Aug. 5 – 8, 2012, pp. 325–328. doi:10.1109/SSP.2012.6319694

[62] J. Barbier, “Statistical Physics And Approximate Message Passing Algorithms for Sparse Linear Estimation Problems in Signal Processing and Coding Theory,” Ph.D. dissertation, École Normale Supérieure, 2015.

[63] S. Som, L. C. Potter, and P. Schniter, “Compressive Imaging using Approximate Message Passing and a Markov-Tree Prior,” in *Forty Fourth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, Pacific Grove, California, USA, Nov. 7 – 10, 2010, pp. 243–247. doi:10.1109/ACSSC.2010.5757509

[64] S. Som and P. Schniter, “Approximate Message Passing for Recovery of Sparse Signals with Markov-Random-Field Support Structure,” in *International Conference on Machine Learning (ICML)*, Bellevue, Washington, USA, Jul. 2, 2011.

[65] E. W. Tramel, A. Dremeau, and F. Krzakala, “Approximate message passing with restricted Boltzmann machine priors,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2016, no. 7, pp. 1–15, Jul. 2016. doi:10.1088/1742-5468/2016/07/073401

[66] P. Schniter and S. Rangan, “Compressive Phase Retrieval via Generalized Approximate Message Passing,” *IEEE Transactions on Signal Processing*, vol. 63, no. 4, pp. 1043–1055, Feb. 2015. doi:10.1109/TSP.2014.2386294

[67] J. Vila, P. Schniter, and J. Meola, “Hyperspectral Image Unmixing via Bilinear Generalized Approximate Message Passing,” in *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XIX, Proceedings of the SPIE*, S. S. Shen and P. E. Lewis, Eds., vol. 8743, Baltimore, Maryland, USA, Apr. 29, 2013, pp. (87 430Y–1)–(87 430Y–9). doi:10.1117/12.2015859

[68] ——, “Hyperspectral Unmixing via Turbo Bilinear Approximate Message Passing,” *IEEE Transactions on Computational Imaging*, vol. 1, no. 3, pp. 143 – 158, Sep. 2015. doi:10.1109/TCI.2015.2465161

[69] J. Vila and P. Schniter, “An Empirical-Bayes Approach to Recovering Linearly Constrained Non-Negative Sparse Signals,” in *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, St. Martin, France, Dec. 15 – 18, 2013, pp. 5–8. doi:10.1109/CAMSAP.2013.6713993

[70] J. Ziniel, L. C. Potter, and P. Schniter, “Tracking and Smoothing of Time-Varying Sparse Signals via Approximate Belief Propagation,” in *Forty Fourth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, Pacific Grove, California, USA, Nov. 7 – 10, 2010, pp. 808–812. doi:10.1109/ACSSC.2010.5757677

[71] J. Ziniel and P. Schniter, “Dynamic Compressive Sensing of Time-Varying Signals Via Approximate Message Passing,” *IEEE Transactions on Signal Processing*, vol. 61, no. 21, pp. 5270–5284, Nov. 2013. doi:10.1109/TSP.2013.2273196

[72] ——, “Efficient Message Passing-Based Inference in the Multiple Measurement Vector Problem,” in *Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, Pacific Grove, California, USA, Nov. 6 – 9, 2011, pp. 1447–1451. doi:10.1109/ACSSC.2011.6190257

[73] ——, “Efficient High-Dimensional Inference in the Multiple Measurement Vector Problem,” *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 340–354, Jan. 2013. doi:10.1109/TSP.2012.2222382

[74] J. Ziniel, P. Schniter, and P. Sederberg, “Binary Linear Classification and Feature Selection via Generalized Approximate Message Passing,” in *48th Annual Conference on Information Sciences and Systems (CISS)*, Princeton, New Jersey, USA, Mar. 19 – 21, 2014, pp. 1–6. doi:10.1109/CISS.2014.6814160

[75] E. Byrne and P. Schniter, “Sparse Multinomial Logistic Regression via Approximate Message Passing,” *IEEE Transactions on Signal Processing*, vol. 64, no. 21, pp. 5485–5498, Nov. 2016. doi:10.1109/TSP.2016.2593691

[76] S. Rangan and A. K. Fletcher, “Iterative Estimation of Constrained Rank-One Matrices in Noise,” in *IEEE International Symposium on Information (ISIT)*, Cambridge, Massachusetts, USA, Jul. 1 – 6, 2012, pp. 1246–1250. doi:10.1109/ISIT.2012.6283056

[77] J. Barbier, C. Schulke, and F. Krzakala, “Approximate message-passing with spatially coupled structured operators, with applications to compressed sensing and sparse superposition codes,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2015, no. 5, p. P05013, May 2015. doi:10.1088/1742-5468/2015/05/P05013

[78] J. Barbier, E. W. Tramel, and F. Krzakala, “Scampi: a robust approximate message-passing framework for compressive imaging,” in *International Meeting on High-Dimensional Data-Driven Science*, ser. Journal of Physics: Conference Series, vol. 699, Kyoto, Japan, Dec. 14 – 17, 2015, pp. 1–13. doi:10.1088/1742-6596/699/1/012013

[79] M. Borgerding, P. Schniter, J. Vila, and S. Rangan, “Generalized Approximate Message Passing for Cosparse Analysis Compressive Sensing,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, South Brisbane, Queensland, Australia, Apr. 19 – 24, 2015, pp. 3756–3760. doi:10.1109/ICASSP.2015.7178673

[80] U. Kamilov, V. K. Goyal, and S. Rangan, “Generalized Approximate Message Passing Estimation from Quantized Samples,” in *4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, San Juan, Puerto Rico, Dec. 13 – 16, 2011, pp. 365–368. doi:10.1109/CAMSAP.2011.6136027

[81] ——, “Optimal Quantization for Compressive Sensing under Message Passing Reconstruction,” in *IEEE International Symposium on Information Theory (ISIT)*, St. Petersburg, Russia, Jul. 31 – Aug. 5, 2011, pp. 459–463. doi:10.1109/ISIT.2011.6034168

[82] K. Sung, B. L. Daniel, and B. A. Hargreaves, “Location Constrained Approximate Message Passing for Compressed Sensing MRI,” *Magnetic Resonance in Medicine*, vol. 70, no. 2, p. 370–381, Aug. 2013. doi:10.1002/mrm.24468