# Approximate Message Passing Algorithms for Generalized Bilinear Inference

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Jason T. Parker, B.S., M.S.

Graduate Program in Department of Electrical and Computer Engineering

The Ohio State University

2014

Dissertation Committee:

Phil Schniter, Advisor

Lee Potter

Emre Ertin

# Abstract

Recent developments in compressive sensing (CS) combined with increasing demands for effective high-dimensional inference techniques across a variety of disciplines have motivated extensive research into algorithms exploiting various notions of parsimony, including sparsity and low-rank constraints. In this dissertation, we extend the generalized approximate message passing (GAMP) approach, originally proposed for high-dimensional generalized-linear regression in the context of CS, to handle several classes of bilinear inference problems. First, we consider a general form of noisy CS where there is uncertainty in the measurement matrix as well as in the measurements. Matrix uncertainty is motivated by practical cases in which there are imperfections or unknown calibration parameters in the signal acquisition hardware. While previous work has focused on analyzing and extending classical CS algorithms like the LASSO and Dantzig selector for this problem setting, we propose a new algorithm called Matrix Uncertain GAMP (MU-GAMP) whose goal is minimization of mean-squared error of the signal estimates in the presence of these uncertainties, without attempting to estimate the uncertain measurement matrix itself. Next, we extend GAMP to the generalized-bilinear case, in which the measurement matrix is estimated jointly with the signals of interest, enabling its application to matrix completion, robust PCA, dictionary learning, and related matrix-factorization problems. We derive this Bilinear GAMP (BiG-AMP) algorithm as an approximation of the sum-product

belief propagation algorithm in the high-dimensional limit, where central limit theorem arguments and Taylor-series approximations apply, and under the assumption of statistically independent matrix entries with known priors. In addition, we propose an adaptive damping mechanism that aids convergence under finite problem sizes, an expectation-maximization (EM)-based method to automatically tune the parameters of the assumed priors, and two rank-selection strategies. We then discuss the specializations of EM-BiG-AMP to the problems of matrix completion, robust PCA, and dictionary learning, and present the results of an extensive empirical study comparing EM-BiG-AMP to state-of-the-art algorithms on each problem. Our numerical results, using both synthetic and real-world datasets, demonstrate that EM-BiG-AMP yields excellent reconstruction accuracy (often best in class) while maintaining competitive runtimes and avoiding the need to tune algorithmic parameters. Finally, we propose a parametric extension known as P-BiG-AMP, which recovers BiG-AMP as a special case, that relaxes the assumption of statistically independent matrix entries by introducing parametric models for the two matrix factors. The resulting algorithm is rigorously justified for random affine parameterizations and constructed to allow its use with an even wider class of non-linear parameterizations, enabling numerous potential applications.

For my mother, who has given so much and asked so little

# Acknowledgments

I would like to thank my advisor Professor Phil Schniter for his patience, wisdom, technical acumen, and support over the last several years. His incredible insight and creative approaches to problem solving made this work possible. I have also benefited tremendously as a programmer, writer, and researcher from his guidance. Professor Lee Potter introduced me to compressed sensing and provided invaluable support as I applied these techniques to problems in radar signal processing. He has been a great friend and mentor throughout this experience. I would also like to thank Emre Ertin for serving on my committee and providing code and data to support my investigations. Andrea Serrani graciously served on my candidacy committee and advised me during the completion of my M.S. degree. I am also grateful to the numerous students in the department who enriched my studies and freely provided their time and talents to advance my research. Subhojit Som, Julie Jackson, Christian Austin, and numerous other members of the compressed sensing reading group provided useful discussions and keen insights. I would also like to specifically thank Justin Ziniel and Jeremy Vila for their generosity with both their time and code.

Beyond OSU, Professor Volkan Cevher at EPFL has been a great collaborator on BiG-AMP. I would also like to thank Professor Sundeep Rangan and the other contributors to the GAMPmatlab project for providing invaluable software tools in

support of this research. I've had the opportunity to work for two excellent supervisors at AFRL during my graduate studies, Bill Baldygo and Jeff Sanders. I am grateful to both for their support of my continued research and patience with the required time commitment. Numerous other colleagues at AFRL, indeed too many to list, have provided friendship, mentorship, and support. I do want to specifically thank Murali Rangaswamy, Braham Himed, and Mike Wicks for their technical leadership. John Scanlan, Ken Schafer, and Shaun Frost have also been dear friends and invaluable teammates. Several colleagues in industry and academia have also provided good advice and technical insights over the years, including Matt Ferrara, Bill Melvin, Margaret Cheney, and many others. This research would not have been possible without the generous support of AFRL, Dr. Arje Nachman at AFOSR, and the National Science Foundation Graduate Research Fellowship.

My family has been incredibly supportive throughout my studies. I'm grateful to my late father for the reverence of learning that he instilled in me at an early age. My mother has provided financial and emotional support throughout. I have never known a more giving or selfless person, and I have no doubt that I owe much of who I am to her. My brother Brian has also been an invaluable companion throughout the years. The monday night crew has kept me sane and made me smile. Finally, I want to thank my wife and dearest friend, Laura Humphrey. She has been there for me from the beginning, and this degree is as much hers as mine. You make it all worth it, and I love you.

# Vita

| | |
|---|---|
| February 18, 1982 | Born - Kettering, OH, USA |
| 2000 | Summer Intern, AFRL/PRSF |
| 2001 | Summer Intern, AFRL/PRTC |
| 2002-2004 | Special Consultant to the Pulsed Detonation Engine Research Facility at AFRL/PRTC, ISSI |
| 2004 | B.S. Electrical Engineering, The Oho State University |
| 2005 | Invited Scholar, AFRL/VACA |
| 2006 | M.S. Electrical Engineering, The Ohio State University |
| 2004-2006, 2008-2009 | National Science Foundation Graduate Research Fellow |
| 2006-present | Research Engineer, Air Force Research Laboratory, Sensors Directorate. |

# Publications

### Research Publications

M. Ferrara, J. T. Parker, and M. Cheney, "Resolution Optimization with Irregularly Sampled Fourier Data," *Inverse Problems*, vol. 29, no. 5, 2013.

J. T. Parker, M. Ferrara, L. C. Potter, "Radar Applications of Sparse Reconstruction and Compressed Sensing," Chapter 5 in *Principles of Modern Radar: Advanced*

*Techniques*, Edited by W. L. Melvin and J. A. Scheer, SciTech Publishing, September 2012.

J. T. Parker and P. Schniter, Bilinear generalized approximate message passing (BiG-AMP) for matrix completion, presented at *Asilomar Conference on Signals, Systems, and Computers*, Nov. 2012.

P. Schniter, J. T. Parker, and V. Cevher, Bilinear generalized approximate message passing (BiG-AMP) for matrix recovery problems, presented at *Information Theory and Applications Workshop*, Feb. 2012.

J. T. Parker, V. Cevher, and P. Schniter, "Compressive Sensing under Matrix Uncertainties: An Approximate Message Passing Approach," *Proceedings of Asilomar Conference on Signals, Systems, and Computers*, November 2011.

J. T. Parker, L. J. Moore, and L. C. Potter, "Resolution and Sidelobe Structure Analysis for RF Tomography," *Proceedings of IEEE Radarcon*, 2011.

L. C. Potter, E. Ertin, J. T. Parker, and M. Cetin, "Sparsity and Compressed Sensing in Radar Imaging," *Proceedings of the IEEE*, vol. 98, pp. 1006-1020, June 2010.

J. T. Parker and L. C. Potter, "A Bayesian Perspective on Sparse Regularization for STAP Post-Processing," *Proceedings of IEEE Radarcon*, 2010.

## Fields of Study

Major Field: Electrical and Computer Engineering

# Table of Contents

**Page**

# List of Tables

# List of Figures

# Chapter 1: Introduction

The origin of the name Compressive Sensing (CS) [1, 2] lies in a particular interpretation of CS algorithms as an approach to signal compression. Many practical systems sample a signal of interest at a rate above the Nyquist rate dictated by its bandwidth, transform the signal to a basis which concentrates the signal's energy into a few large entries, and achieve compression by storing the values and locations of only these dominant coefficients. JPEG2000 is an excellent example of processing in this vein, relying on a wavelet transformation to sparsify natural images. Since the signal is eventually encoded with only a few coefficients, it seems natural to ask if fewer measurements could have been taken in the first place. Under various technical conditions, CS accomplishes this goal by combining a reduced number of, typically randomized, measurements with nonlinear reconstruction algorithms and known sparsifying transformations. In particular, CS algorithms typically consider linear regression problems of the form $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}$, with noise $\boldsymbol{w}$, a sparse or compressible signal $\boldsymbol{x} \in \mathbb{R}^N$, and an under-determined linear operator $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ with $M < N$.

Over the past several years, work on CS has spurred research into several classes of algorithms for under-determined linear regression, including methods in convex optimization, greedy techniques, and various Bayesian approaches. Of particular interest

here is Approximate Message Passing (AMP) [3–5] and the subsequent Generalized AMP (GAMP) [6], which offers state-of-the-art performance for linear CS problems, significant flexibility in the assumed prior models for the signal of interest $\boldsymbol{x}$, a generalized measurement model that supports non-Gaussian noise and even non-linear measurements, and low implementation complexity. These methods, like the vast majority of other CS techniques, treat the measurement matrix $\boldsymbol{A}$ as fixed and known. In this dissertation, motivated by the success of AMP methods for solving *linear* problems in CS, we consider extensions of the AMP framework to handle various *bilinear* inference problems in which $\boldsymbol{A}$ is partially or even completely unknown.

This dissertation is organized as follows. Chapter 2 reviews prior work on GAMP and includes a detailed derivation of the algorithm. Various modifications required for its practical implementation are also described, along with the simplifications required to reduce GAMP to the original AMP. While GAMP itself does not represent a contribution of this dissertation, we have made notable contributions to its publicly available implementation, which we have also extended to include the novel algorithms presented in subsequent chapters. Next, Chapter 3 summarizes our first contribution, an extension of GAMP which models the entries of $\boldsymbol{A}$ as independent random variables with known means and variances in an effort to recover $\boldsymbol{x}$ more accurately in the presence of an uncertain measurement matrix. Empirically, we show that this Matrix Uncertain GAMP (MU-GAMP) [7] approach performs near oracle bounds. We also present a simple analysis showing that, for suitably large systems, it suffices to treat uniform matrix uncertainty as additive white Gaussian noise. While MU-GAMP does not estimate $\boldsymbol{A}$ directly, we show that it can be applied in an alternating fashion,

referred to as A-MU-GAMP, to learn both the signal vector and the measurement matrix when $\boldsymbol{A}$ can be written as an affine combination of known matrices.

In Chapter 4, we pursue a more flexible approach which *jointly* estimates $\boldsymbol{A}$ and matrix-valued $\boldsymbol{X}$ by placing separable priors on the elements of both matrix factors. This *bilinear* inference algorithm is referred to as Bilinear GAMP (BiG-AMP) [8, 9]. We present several special-case simplifications of the algorithm that offer reduced computational complexity. In addition, an adaptive damping mechanism, expectation maximization (EM) based tuning of the prior parameters, and two methods for selecting the rank $N$ of the product $\boldsymbol{AX}$ are presented. The resulting algorithm is then specialized for matrix completion, robust principal components anlaysis, and dictionary learning. A detailed empirical study shows that BiG-AMP yields an excellent combination of estimation accuracy and runtime when compared to existing state-of-the-art algorithms for each application.

Our final contribution is described in Chapter 5, where we develop a parametric extension of BiG-AMP. This P-BiG-AMP algorithm [10] handles the case where $\boldsymbol{A}$ and $\boldsymbol{X}$ are described by known parametric models and seeks to jointly estimate the parameters of these models, rather than the matrices themselves. This approach reduces to BiG-AMP for "trivial" deterministic parameterizations, but can also handle much more general settings where the effective priors on $\boldsymbol{A}$ and $\boldsymbol{X}$ are non-separable, e.g., a measurement system with a small number of calibration parameters. In the interest of generality, we carry out the derivation for possibly non-linear parameterizations, although certain steps in the derivation are rigorously justified only in the case of random affine parameterizations. Practical implementation issues for P-BiG-AMP are also addressed, along with numerical examples to demonstrate the technique's

effectiveness. Finally, we offer conclusions and a summary of possible future work in Chapter 6.

# Chapter 2: Generalized Approximate Message Passing

## 2.1 Overview

In this chapter, we provide a detailed derivation of the Generalized Approximate Message Passing (GAMP) algorithm from [6] and describe several issues surrounding its practical implementation. GAMP is a generalization of Donoho/Maleki/Montanari's AMP [3–5], where the latter handled only Gaussian output channels. The GAMP analysis closely follows the AMP analysis of Bayati/Montanari [11]. GAMP is also closely related to Guo/Wang's relaxed BP [12,13], but admits a rigorous analysis with dense matrices $\boldsymbol{A}$, as well as a somewhat simpler implementation. Rangan's work [6] provides asymptotic analysis that shows that, for i.i.d Gaussian $\boldsymbol{A}$, as $\lim_{N\to\infty} \frac{N}{M(N)} = \beta$, the empirical distributions of GAMP estimates converge (at all iterations) to distributions predicted by a state evolution (SE) formalism. Moreover, the SE equations are shown to coincide with those derived using the non-rigorous replica method from statistical physics, as well as with those derived using sparse-matrix assumptions. Mismatched statistics are also considered. We refer the reader to [6] for the details of these analyses, as our focus here will be on the derivation and implementation of the algorithm. The derivation is based primarily on Taylor Series and Central Limit Theorem (CLT) arguments and introduces several

techniques used in the novel derivations carried out in subsequent chapters. While Rangan provides versions of GAMP for both minimum mean squared error (MMSE) and maximum a posteriori (MAP) estimation, here we restrict our attention to the MMSE case.

The chapter is organized as follows: in Section 2.2, we set up the problem of interest for GAMP. We then introduce the sum product algorithm in Section 2.3 and apply it to derive MMSE GAMP in Section 2.4. Next, example channel models along with some practical convergence issues are addressed in Section 2.5 to Section 2.7. Finally, we describe the cost function used for step acceptance in practical implementations of GAMP in Section 2.8 and describe the connection between GAMP and AMP in Section 2.9.

## 2.2 Problem Setup

We denote the signal vector of interest as $\boldsymbol{x} \in \mathbb{R}^N$, drawn from the arbitrary separable pdf

$$p(\boldsymbol{x} \,|\, \boldsymbol{q}) = \prod_j p_{X|Q}(x_j \,|\, q_j), \tag{2.1}$$

where $\{q_j\}$ are known parameters. Given the known measurement matrix $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ with entries $a_{ij} = [\boldsymbol{A}]_{i,j}$, sometimes with $M < N$, we define the *noiseless* measurements (in practice unknown) as $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{x} \in \mathbb{R}^M$. The vector of known noisy measurements $\boldsymbol{y} \in \mathbb{R}^M$ is related to $\boldsymbol{z}$ via the arbitrary separable "output channel"

$$p(\boldsymbol{y} \,|\, \boldsymbol{z}) = \prod_i p_{Y|Z}(y_i \,|\, z_i). \tag{2.2}$$

Our goal is to estimate $\boldsymbol{x}$ from $\boldsymbol{y}$ using the MMSE criterion.

## 2.3 Review of the Sum-Product Algorithm

The sum-product algorithm [14, 15] is a form of belief propagation (BP) that attempts to iteratively estimate the marginals of the posterior $p(\boldsymbol{x} \,|\, \boldsymbol{y}, \boldsymbol{q})$:

$$\begin{aligned} p(\boldsymbol{x} \,|\, \boldsymbol{y}, \boldsymbol{q}) &= \frac{p(\boldsymbol{y} \,|\, \boldsymbol{x}) p(\boldsymbol{x} \,|\, \boldsymbol{q})}{p(\boldsymbol{y} \,|\, \boldsymbol{q})} \propto p(\boldsymbol{y} \,|\, \boldsymbol{x}) p(\boldsymbol{x} \,|\, \boldsymbol{q}) \\ &= \prod_{i=1}^{M} p_{Y|Z}(y_i \,|\, [\boldsymbol{A}\boldsymbol{x}]_i) \prod_{j=1}^{N} p_{X|Q}(x_j \,|\, q_j). \end{aligned} \tag{2.3}$$

In BP, messages are passed between nodes of the factor graph of $p(\boldsymbol{x} \,|\, \boldsymbol{y}, \boldsymbol{q})$, illustrated in Fig. 2.1. There, the factors of $p(\boldsymbol{x} \,|\, \boldsymbol{y}, \boldsymbol{q})$ are represented by "function nodes" depicted as black squares and random variables in $p(\boldsymbol{x} \,|\, \boldsymbol{y}, \boldsymbol{q})$ are represented by "variable nodes," depicted as white circles. Each variable (node) is connected to every factor (node) in which it appears. Note that since $\{y_i\}_{i=1}^{M}$ and $\{q_j\}_{j=1}^{N}$ are known, they are treated as pdf parameters and not as random variables.

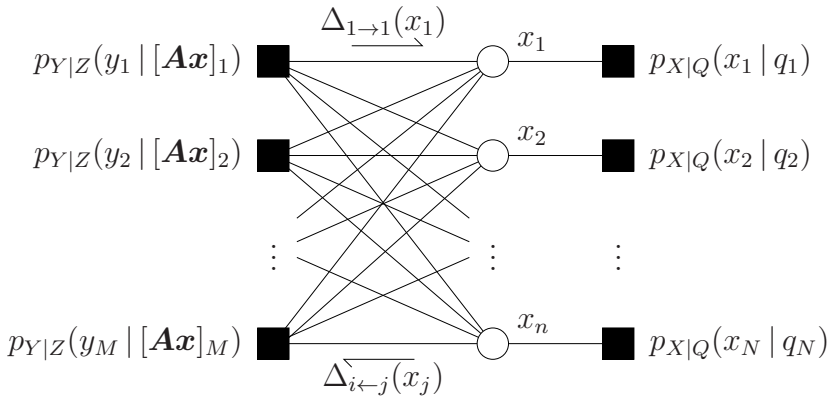


Figure 2.1: The factor graph for generalized linear inference.

In our formulation of the sum product algorithm [14, 15], messages take the form of log-pdfs, with arbitrary constant offsets. For example, the message $\Delta_{i\to j}(x_j)$ corresponds to the pdf $\frac{1}{Z}\exp(\Delta_{i\to j}(x_j))$, where $Z \triangleq \int_{x_j}\exp(\Delta_{i\to j}(x_j))$ is a necessary scaling factor. The sum-product algorithm estimates the posterior pdf of a given variable as the product of exponentiated messages entering that variable node (after appropriate scaling). For example, on the graph in Fig. 2.1, the estimate of $p(x_j \mid \boldsymbol{y}, \boldsymbol{q})$ takes the form of $\frac{1}{Z}\exp\big(\Delta_j(x_j)\big)$, where

$$\Delta_j(x_j) \;\; = \;\; \mathsf{const} + \log p_{X|Q}(x_j|q_j) + \sum_{i=1}^{m} \Delta_{i\to j}(x_j), \tag{2.4}$$

and where $\Delta_{i\to j}(x_j)$ denotes the message from the $i^{th}$ function node (on the left) to the $j^{th}$ variable node (on the right). We use $\mathsf{const}$ to denote an arbitrary constant offset. Without loops in the graph, BP yields exact posteriors with one round of message passing. With loops, exact inference is generally NP hard, but approximate inference via BP can perform very well, such as in the case described here. We now apply the sum product algorithm to the graph in Fig. 2.1 to arrive at update rules for the messages $\Delta_{i\to j}(x_j)$ and $\Delta_{i\leftarrow j}(x_j)$.

First, the outgoing message from a factor node on a given branch is the log of the integrated product of the local function and all exponentiated incoming messages on other branches. Applying this procedure to the function-to-variable messages $\Delta_{i\to j}(x_j)$, we obtain

$$\Delta_{i\to j}(x_j) \;\; = \;\; \mathsf{const} + \log \int_{\{x_r\}_{r\neq j}} p_{Y|Z}(y_i \mid [\boldsymbol{A}\boldsymbol{x}]_i) \;\; \prod_{r\neq j} \exp\big(\Delta_{i\leftarrow r}(x_r)\big). \tag{2.5}$$

The calculation of this message is depicted graphically in Fig. 2.2. For variable nodes, the outgoing message on a given branch is the sum of all incoming messages on other

branches, which allows us to compute the variable-to-function messages $\Delta_{i\leftarrow j}(x_j)$ as

$$\Delta_{i\leftarrow j}(x_j) \;=\; \mathsf{const} + \log p_{X|Q}(x_j \,|\, q_j) + \sum_{l\neq i} \Delta_{l\to j}(x_j). \tag{2.6}$$

A graphical depiction of this message calculation is provided in Fig. 2.3. For the factor graph in Fig. 2.1, exact implementation of the sum-product algorithm has complexity exponential in $N$, motivating the simpler scheme described in the sequel.

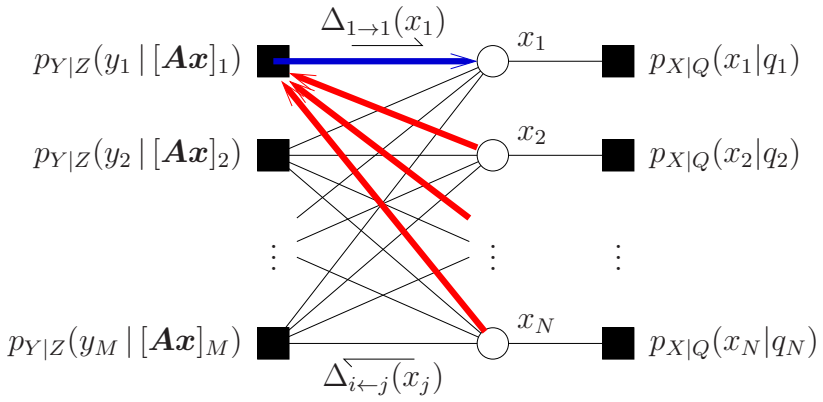


Figure 2.2: A graphical depiction of the portion of the factor graph used to compute $\Delta_{i\to j}(x_j)$. The red arrows represent the incoming messages used in the calculation, while the blue arrow marks the resulting message.

## 2.4 GAMP for the MMSE Case

The sum-product algorithm is now approximated using central limit theorem (CLT) and Taylor-series ideas, yielding the MMSE version of the GAMP algorithm.

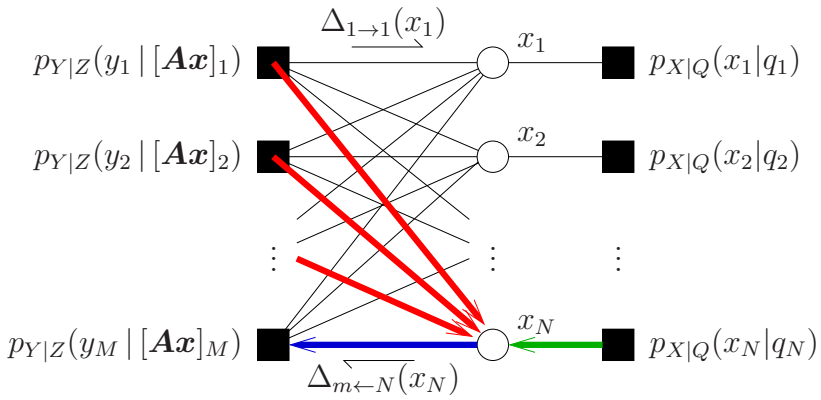


Figure 2.3: A graphical depiction of the portion of the factor graph used to compute $\Delta_{i\leftarrow j}(x_j)$. The red and green arrows represent the incoming messages used in the calculation, while the blue arrow marks the resulting message. The message from the prior node $p_{X|Q}(x_N|q_N)$ is shown in green to emphasize that this message is known and does not need to be recomputed for each iteration.

The following means and variances are used in the sequel:

$$\widehat{x}_j(t) \triangleq \mathrm{E}\{x_j \,|\, \Delta_j(t,.)\} \tag{2.7}$$

$$\mu_j^x(t) \triangleq \mathrm{var}\{x_j \,|\, \Delta_j(t,.)\} \tag{2.8}$$

$$\widehat{x}_{ij}(t) \triangleq \mathrm{E}\{x_j \,|\, \Delta_{i\leftarrow j}(t,.)\} \tag{2.9}$$

$$\mu_{ij}^x(t) \triangleq \mathrm{var}\{x_j \,|\, \Delta_{i\leftarrow j}(t,.)\}, \tag{2.10}$$

where in (2.7)-(2.10) it is assumed that $x_j \sim \frac{1}{Z}\exp(\Delta(t,.))$ for $Z \triangleq \int_{x_j}\exp(\Delta(t,x_j))$.

In the sequel, we approximate the updates (2.4)-(2.6) using the AMP framework. Our approximations will be guided by analyzing the algorithm in the large system limit, where $\delta \triangleq M/N$ is held constant as $N \to \infty$. Similar to the derivations in [6, 16,17], we shall assume that $x_j$ is $O(1)$ and $a_{ij}$ is $O(1/\sqrt{N})$ and drawn randomly from a zero mean distribution, which implies that $z_i$ is $O(1)$. Based on these assumptions and our subsequent derivation, Table 2.1 summarizes how several variables used by

| | | | |
|---|---|---|---|
| $\widehat{x}_{ij}(t)$ | $O(1)$ | $\mu^x_{ij}(t)$ | $O(1)$ |
| $\widehat{x}_j(t)$ | $O(1)$ | $\mu^x_j(t)$ | $O(1)$ |
| $\widehat{r}_{ij}(t)$ | $O(1)$ | $\mu^r_{ij}(t)$ | $O(1)$ |
| $\widehat{r}_j(t)$ | $O(1)$ | $\mu^r_j(t)$ | $O(1)$ |
| $\widehat{z}_i(t)$ | $O(1)$ | $\mu^z_i(t)$ | $O(1)$ |
| $\widehat{p}_i(t)$ | $O(1)$ | $\mu^p_i(t)$ | $O(1)$ |
| $\widehat{s}_i(t)$ | $O(1)$ | $\mu^s_i(t)$ | $O(1)$ |
| $\widehat{x}_{ij}(t) - \widehat{x}_j(t)$ | | | $O(1/\sqrt{N})$ |
| $\mu^r_{ij}(t) - \mu^r_j(t)$ | | | $O(1/N)$ |
| $\widehat{p}_{ij}(t) - \widehat{p}_i(t)$ | | | $O(1/\sqrt{N})$ |
| $\mu^p_{ij}(t) - \mu^p_i(t)$ | | | $O(1/N)$ |

Table 2.1: GAMP variable scalings in the large system limit.

GAMP, most of which will be defined in the sequel, are assumed to scale relative to $N$ in the large system limit.

### 2.4.1 Messages emitted by the function (or "output") nodes

Rule (2.5) of the sum-product algorithm yields the time-$t$ message

$$\begin{aligned}
&\Delta_{i\to j}(t, x_j) \\
&\quad = \quad \mathsf{const} + \log \int_{\{x_r\}_{r\neq j}} p_{Y|Z}\Big(y_i \,\Big|\, \underbrace{a_{ij}x_j + \sum_{r\neq j} a_{ir}x_r}_{\triangleq\, z_i}\Big) \prod_{r\neq j} e^{\Delta_{i\leftarrow r}(t,x_r)}. \qquad (2.11)
\end{aligned}$$

For large $N$, the CLT motivates the treatment of $z_i$ as conditionally Gaussian, i.e.,

$$z_i|x_j \quad \sim \quad \mathcal{N}\big(a_{ij}x_j + \widehat{p}_{ij}(t), \mu^p_{ij}(t)\big), \qquad (2.12)$$

where

$$\widehat{p}_{ij}(t) \quad \triangleq \quad \sum_{r\neq j} a_{ir}\widehat{x}_{ir}(t) \qquad (2.13)$$

$$\mu^p_{ij}(t) \quad \triangleq \quad \sum_{r\neq j} a^2_{ir}\mu^x_{ir}(t). \qquad (2.14)$$

Using this CLT approximation,

$$\Delta_{i\to j}(t,x_j) \;\approx\; \mathsf{const} + \underbrace{\log \int_{z_i} p_{Y|Z}(y_i \,|\, z_i)\, \mathcal{N}\big(z_i; a_{ij}x_j + \widehat{p}_{ij}(t), \mu^p_{ij}(t)\big)}_{\triangleq\, H\left(a_{ij}x_j + \widehat{p}_{ij}(t), y_i, \mu^p_{ij}(t)\right)} \tag{2.15}$$

The related quantities

$$\widehat{p}_i(t) \;\triangleq\; \sum_j a_{ij}\widehat{x}_{ij}(t) \tag{2.16}$$

$$\mu^p_i(t) \;\triangleq\; \sum_j a^2_{ij}\mu^x_{ij}(t) \tag{2.17}$$

can be plugged into (2.15) to get

$$\Delta_{i\to j}(t,x_j) \;\approx\; \mathsf{const} + H\big(a_{ij}(x_j - \widehat{x}_{ij}(t)) + \widehat{p}_i(t), y_i, \mu^p_{ij}(t)\big) \tag{2.18}$$

$$= \;\mathsf{const} + H\big(a_{ij}(x_j - \widehat{x}_j(t)) + \widehat{p}_i(t) + O(1/N), \ldots$$

$$y_i, \mu^p_i(t) + O(1/N)\big), \tag{2.19}$$

where in (2.19) we have recognized, recalling Table 2.1, that $a_{ij}[\widehat{x}_j(t) - \widehat{x}_{ij}(t)]$ is $O(1/N)$. The difference between $\mu^p_{ij}(t)$ and $\mu^p_i(t)$ is also $O(1/N)$, since they differ by a single term that is scaled by $a^2_{ij}$. Notice that the remaining terms in both function arguments are $O(1)$. Applying a Taylor-series approximation to (2.19), one finds

$$\begin{aligned}\Delta_{i\to j}(t,x_j) \;\approx\; & \mathsf{const} + H\big(\widehat{p}_i(t), y_i, \mu^p_i(t)\big) \\ & + a_{ij}(x_j - \widehat{x}_j(t))H'\big(\widehat{p}_i(t), y_i, \mu^p_i(t)\big) \\ & + \tfrac{1}{2}a^2_{ij}(x_j - \widehat{x}_j(t))^2 H''\big(\widehat{p}_i(t), y_i, \mu^p_i(t)\big),\end{aligned} \tag{2.20}$$

where $H'(\cdot,\cdot,\cdot)$ denotes the derivative of $H(\cdot,\cdot,\cdot)$ with respect to the first argument, and $H''(\cdot,\cdot,\cdot)$ the second derivative. A sum containing $O(N)$ of the $\Delta_{i\to j}(t,x_j)$ messages is used to compute $\Delta_{i\leftarrow j}(x_j)$. Thus, we have only retained terms in the expansion that are $O(1/N)$ or larger. In particular, we have dropped the $O(1/N)$

perturbations of the $H'(\cdot,\cdot,\cdot)$ and $H''(\cdot,\cdot,\cdot)$ function arguments[1] and the cubic[2] and higher order terms in the Taylor series. Defining

$$\widehat{s}_i(t) \triangleq H'\big(\widehat{p}_i(t), y_i, \mu_i^p(t)\big) \tag{2.21}$$

$$\mu_i^s(t) \triangleq -H''\big(\widehat{p}_i(t), y_i, \mu_i^p(t)\big), \tag{2.22}$$

and noting that $H\big(\widehat{p}_i(t), y_i, \mu_i^p(t)\big)$ is constant w.r.t $x_j$, (2.20) becomes

$$\Delta_{i\to j}(t, x_j) \approx \mathsf{const} + a_{ij}(x_j - \widehat{x}_j(t))\widehat{s}_i(t) - \tfrac{1}{2}a_{ij}^2\big(x_j - \widehat{x}_j(t)\big)^2\mu_i^s(t) \tag{2.23}$$

$$= \mathsf{const} + \big[\widehat{s}_i(t)a_{ij} + \mu_i^s(t)a_{ij}^2\widehat{x}_j(t)\big]x_j - \tfrac{1}{2}\mu_i^s(t)a_{ij}^2x_j^2. \tag{2.24}$$

In essence, the pdf $\frac{1}{Z}\exp(\Delta_{i\to j}(t, x_j))$ has been approximated as Gaussian, even though $p_{Y|Z}(.|.)$ in (2.15) may be non-Gaussian.

As shown below, the quantities in (2.21) and (2.22) can be simplified. From

$$H(\widehat{p}, y, \mu^p) \triangleq \log \int p_{Y|Z}(y \mid z)\,\mathcal{N}(z; \widehat{p}, \mu^p)dz, \tag{2.25}$$

it can be seen that

$$\begin{aligned}
&H'(\widehat{p}, y, \mu^p) \\
&= \frac{\partial}{\partial\widehat{p}}\log\int p_{Y|Z}(y \mid z)\frac{1}{\sqrt{2\pi\mu^p}}\exp\Big(-\frac{1}{2\mu^p}(z-\widehat{p})^2\Big)dz && (2.26)\\
&= \frac{\partial}{\partial\widehat{p}}\Big\{\log\frac{1}{\sqrt{2\pi\mu^p}} + \log\int_z \exp\Big(\log p_{Y|Z}(y \mid z) - \frac{1}{2\mu^p}(z-\widehat{p})^2\Big)dz\Big\} && (2.27)\\
&= \frac{\partial}{\partial\widehat{p}}\Big\{-\frac{\widehat{p}^2}{2\mu^p} + \log\int \exp\Big(\log p_{Y|Z}(y \mid z) - \frac{z^2}{2\mu^p} + \frac{\widehat{p}z}{\mu^p}\Big)dz\Big\} && (2.28)\\
&= -\frac{\widehat{p}}{\mu^p} + \frac{\partial}{\partial\widehat{p}}\log\Big[\mu^p\int\exp\big(\phi(u)+\widehat{p}u\big)du\Big] \ \text{ via } u \triangleq \frac{z}{\mu^p} && (2.29)\\
&= -\frac{\widehat{p}}{\mu^p} + \frac{\partial}{\partial\widehat{p}}\log\int\exp\big(\phi(u)+\widehat{p}u\big)du && (2.30)
\end{aligned}$$

[1]A further Taylor series expansion of these terms reveals that these perturbations can be neglected, since the coefficients premultiplying them are $O(1/\sqrt{N})$ or smaller.

[2]Notice that the cubic term is scaled by $a_{ij}^3$, which is $O(1/N^{3/2})$.

for appropriate $\phi(\cdot)$. Now, for $Z(\widehat{p}) \triangleq \int \exp\big(\phi(u) + \widehat{p}u\big)du$, simple calculus yields

$$\frac{\partial}{\partial \widehat{p}} \log Z(\widehat{p}) \;=\; \mathrm{E}\{u \,|\, \widehat{p}\} \;\;\text{with}\;\; p_{U|P}(u \,|\, \widehat{p}) = \tfrac{\exp(\phi(u)+\widehat{p}u)}{Z(\widehat{p})} \tag{2.31}$$

$$\frac{\partial^2}{\partial \widehat{p}^2} \log Z(\widehat{p}) \;=\; \mathrm{var}\{u \,|\, \widehat{p}\} \;\;\text{with}\;\; p_{U|P}(u \,|\, \widehat{p}) = \tfrac{\exp(\phi(u)+\widehat{p}u)}{Z(\widehat{p})}. \tag{2.32}$$

Thus, from (2.30) and (2.31), it follows that

$$H'(\widehat{p}, y, \mu^p) \;=\; -\frac{\widehat{p}}{\mu^p} + \int u \frac{\exp\big(\phi(u) + \widehat{p}u\big)}{Z(\widehat{p})} du \tag{2.33}$$

$$\;=\; -\frac{\widehat{p}}{\mu^p} + \int \frac{z}{\mu^p} \frac{\exp\big(\log p_{Y|Z}(y|z) - \frac{z^2}{2\mu^p} + \frac{z\widehat{p}}{\mu^p}\big)}{Z(\widehat{p})} \frac{dz}{\mu^p} \;\;\text{via}\; u \triangleq \frac{z}{\mu^p} \tag{2.34}$$

$$\;=\; -\frac{\widehat{p}}{\mu^p} + \frac{1}{\mu^p} \int z \frac{\exp\big(\log p_{Y|Z}(y|z) - \frac{1}{2\mu^p}(z-\widehat{p})^2\big)}{\mu^p Z(\widehat{p}) \exp(-\frac{\widehat{p}^2}{2\mu^p})} dz \tag{2.35}$$

$$\;=\; -\frac{\widehat{p}}{\mu^p} + \frac{1}{\mu^p} \int z \frac{p_{Y|Z}(y|z)\mathcal{N}(z; \widehat{p}, \mu^p)}{\int p_{Y|Z}(y|\overline{z})\mathcal{N}(\overline{z}; \widehat{p}, \mu^p) d\overline{z}} dz \tag{2.36}$$

$$\;=\; \frac{1}{\mu^p}\big(\,\mathrm{E}\{z \,|\, y, \widehat{p}; \mu^p\} - \widehat{p}\big), \tag{2.37}$$

where $p_{Z|Y,P}(z \,|\, y, \widehat{p}; \mu^p) \;\propto\; p_{Y|Z}(y|z)\mathcal{N}(z; \widehat{p}, \mu^p)$. This notation invokes the interpretation that GAMP calculates the approximate prior $p_{Z|P}(z_i|\widehat{p}_i(t); \mu_i^p(t)) = \mathcal{N}(z_i; \widehat{p}_i(t), \mu_i^p(t))$ and combines it with the likelihood $p_{Y|Z}(y_i \,|\, z_i)$ to form the resulting posterior $p_{Z|Y,P}(z_i \,|\, y_i, \widehat{p}_i(t); \mu_i^p(t))$, which is an approximation to the true marginal posterior $p_{Z|Y}(z_i \,|\, y_i)$.

Similarly, from (2.30) and (2.32),

$$
\begin{aligned}
&-H''(\widehat{p}, y, \mu^p) \\
&= \frac{\partial}{\partial \widehat{p}}\Big\{ - H'(\widehat{p}, y, \mu^p)\Big\} \;=\; \frac{\partial}{\partial \widehat{p}}\Big\{ \frac{\widehat{p}}{\mu^p} - \frac{\partial}{\partial \widehat{p}} \log Z(\widehat{p})\Big\} && (2.38)\\
&= \frac{1}{\mu^p} - \mathrm{var}\{u \,|\, \widehat{p}\} && (2.39)\\
&= \frac{1}{\mu^p} - \int \big(u - \mathrm{E}\{u \,|\, p\}\big)^2 \, \frac{\exp\big(\phi(u) + \widehat{p}u\big)}{Z(\widehat{p})} du && (2.40)\\
&= \frac{1}{\mu^p} - \int \Big(\frac{z}{\mu^p} - \frac{\mathrm{E}\{z \,|\, y, \widehat{p}; \mu^p\}}{\mu^p}\Big)^2 \, \frac{\exp\big(\log p_{Y|Z}(y|z) - \frac{z^2}{2\mu^p} + \frac{z\widehat{p}}{\mu^p}\big)}{Z(\widehat{p})} \frac{dz}{\mu^p} && (2.41)\\
&= \frac{1}{\mu^p} - \frac{1}{(\mu^p)^2} \int \big(z - \mathrm{E}\{z \,|\, y, \widehat{p}; \mu^p\}\big)^2 \frac{p_{Y|Z}(y|z)\mathcal{N}(z; \widehat{p}, \mu^p)}{\int p_{Y|Z}(y|\overline{z})\mathcal{N}(\overline{z}; \widehat{p}, \mu^p) d\overline{z}} dz && (2.42)\\
&= \frac{1}{\mu^p}\Big(1 - \frac{\mathrm{var}\{z \,|\, y, \widehat{p}; \mu^p\}}{\mu^p}\Big). && (2.43)
\end{aligned}
$$

We define the moments of the time-$t$ approximated marginal posterior as

$$\widehat{z}_i(t) = \mathrm{E}\{z_i \,|\, y_i, \widehat{p}_i(t); \mu_i^p(t)\} \tag{2.44}$$

$$\mu_i^z(t) = \mathrm{var}\{z_i \,|\, y_i, \widehat{p}_i(t); \mu_i^p(t)\}, \tag{2.45}$$

which can be exported for use elsewhere as needed, e.g., in EM learning of likelihood parameters.

### 2.4.2 Messages emitted by the variable (or "input") nodes

Rule (2.6) of the sum-product algorithm yields

$$
\begin{aligned}
&\Delta_{i\leftarrow j}(t{+}1, x_j) \\
&\quad = \quad \mathsf{const} + \log p_{X|Q}(x_j \,|\, q_j) + \sum_{l\neq i} \Delta_{l\to j}(t, x_j) && (2.46)\\
&\quad \approx \quad \mathsf{const} + \log p_{X|Q}(x_j \,|\, q_j) + \sum_{l\neq i} \Big( \big[\widehat{s}_l(t) a_{lj} + \mu_l^s(t) a_{lj}^2 \widehat{x}_j(t)\big] x_j - \tfrac{1}{2}\mu_l^s(t) a_{lj}^2 x_j^2 \Big) && (2.47)\\
&\quad = \quad \mathsf{const} + \log p_{X|Q}(x_j \,|\, q_j) \\
&\qquad - \frac{1}{2\big(\sum_{l\neq i} a_{lj}^2 \mu_l^s(t)\big)^{-1}} \left( x_j - \frac{\sum_{l\neq i} \big(a_{lj}\widehat{s}_l(t) + a_{lj}^2 \mu_l^s(t) \widehat{x}_j(t)\big)}{\sum_{l\neq i} a_{lj}^2 \mu_l^s(t)} \right)^2 && (2.48)\\
&\quad = \quad \mathsf{const} + \log p_{X|Q}(x_j \,|\, q_j) - \frac{1}{2\mu_{ij}^r(t)} \big(x_j - \widehat{r}_{ij}(t)\big)^2 && (2.49)
\end{aligned}
$$

for

$$
\begin{aligned}
\mu_{ij}^r(t) \quad &\triangleq \quad \Big( \sum_{l\neq i} a_{lj}^2 \mu_l^s(t) \Big)^{-1} && (2.50)\\
\widehat{r}_{ij}(t) \quad &\triangleq \quad \widehat{x}_j(t) + \mu_{ij}^r(t) \sum_{l\neq i} a_{lj} \widehat{s}_l(t), && (2.51)
\end{aligned}
$$

where both quantities are $O(1)$. The quantities $\widehat{x}_{ij}(t)$ and $\mu_{ij}^x$ from (2.9) and (2.10) can now be updated using the message approximation (2.49). In particular, (2.49) implies

$$
\widehat{x}_{ij}(t{+}1) \quad \triangleq \quad \mathrm{E}\{x_j \,|\, \Delta_{i\leftarrow j}(t{+}1)\} \quad \approx \quad \underbrace{\frac{\int_x x\, p_{X|Q}(x|q_j)\, \mathcal{N}\big(x; \widehat{r}_{ij}(t), \mu_{ij}^r(t)\big)}{\int_x p_{X|Q}(x|q_j)\, \mathcal{N}\big(x; \widehat{r}_{ij}(t), \mu_{ij}^r(t)\big)}}_{\triangleq\, g_{\mathsf{in}}\left(\widehat{r}_{ij}(t), q_j, \mu_{ij}^r(t)\right)} . \qquad (2.52)
$$

Furthermore, as shown below, (2.52) implies that

$$
\mu_{ij}^x(t{+}1) \quad \triangleq \quad \mathrm{var}\{x_j \,|\, \Delta_{i\leftarrow j}(t{+}1)\} \quad \approx \quad g_{\mathsf{in}}'\big(\widehat{r}_{ij}(t), q_j, \mu_{ij}^r(t)\big)\, \mu_{ij}^r(t), \qquad (2.53)
$$

where $g_{\mathsf{in}}'(\cdot,\cdot,\cdot)$ denotes the derivative of $g_{\mathsf{in}}(\cdot,\cdot,\cdot)$ w.r.t the first argument. As justification for (2.53), we define

$$
G(\widehat{r}, q, \mu^r) \quad \triangleq \quad \log \int p_{X|Q}(x \,|\, q)\, \mathcal{N}(x; \widehat{r}, \mu^r) dx, \qquad (2.54)
$$

which has a very similar form to $H(\widehat{p}, y, \mu^p)$ in (2.25). In fact, following the same steps as before, one finds that

$$G'(\widehat{r}, q, \mu^r) \;=\; \frac{1}{\mu^r}\Big(\,\mathrm{E}\{x \,|\, q, \widehat{r}; \mu^r\} - \widehat{r}\Big) \tag{2.55}$$

$$G''(\widehat{r}, q, \mu^r) \;=\; \frac{1}{\mu^r}\Big(\frac{\mathrm{var}\{x \,|\, q, \widehat{r}; \mu^r\}}{\mu^r} - 1\Big), \tag{2.56}$$

where mean and variance are computed using $p_{X|Q,R}(x|q, \widehat{r}; \mu^r) \propto p_{X|Q}(x \,|\, q)\,\mathcal{N}(x; \widehat{r}, \mu^r)$. This notation invokes the interpretation that GAMP calculates the approximate likelihood $p_{R|X}(\widehat{r}_j(t) \,|\, x_j; \mu_j^r(t))$, which is then combined with the (assumed) prior $p_{X|Q}(x_j \,|\, q_j)$ to form the resulting posterior $p_{X|Q,R}(x_j \,|\, q_j, \widehat{r}_j(t); \mu_j^r(t))$, which is GAMP's iteration-$t$ approximation to the true marginal posterior $p_{X|Y}(x_j \,|\, \boldsymbol{y})$. Then, from (2.52) and (2.55), it follows that

$$g_{\mathsf{in}}(\widehat{r}, q, \mu^r) \;=\; \widehat{r} + \mu^r G'(\widehat{r}, q, \mu^r), \tag{2.57}$$

after which differentiating (2.57) w.r.t $\widehat{r}$ and plugging in (2.56) yields

$$g'_{\mathsf{in}}(\widehat{r}, q, \mu^r) \;=\; 1 + \mu^r G''(\widehat{r}, q, \mu^r) \;=\; \frac{\mathrm{var}\{x \,|\, q, \widehat{r}; \mu^r\}}{\mu^r}, \tag{2.58}$$

which establishes (2.53).

The mean $\widehat{x}_{ij}(t+1)$ is now further approximated using the $O(1)$ quantities

$$\mu_j^r(t) \;\triangleq\; \Big(\sum_i a_{ij}^2 \mu_i^s(t)\Big)^{-1} \tag{2.59}$$

$$\widehat{r}_j(t) \;\triangleq\; \widehat{x}_j(t) + \mu_j^r(t) \sum_i a_{ij}\widehat{s}_i(t). \tag{2.60}$$

Comparison with (2.50) reveals that $\mu_{ij}^r(t) - \mu_j^r(t)$ is $O(1/N)$, as indicated in Table 2.1.

From (2.52), the approximation $\mu_{ij}^r(t) \approx \mu_j^r(t)$ yields

$$\widehat{x}_{ij}(t+1) \;\approx\; g_{\mathsf{in}}\big(\widehat{r}_j(t) - a_{ij}\widehat{s}_i(t)\mu_j^r(t), q_j, \mu_j^r(t)\big) \tag{2.61}$$

$$\;\approx\; \underbrace{g_{\mathsf{in}}\big(\widehat{r}_j(t), q_j, \mu_j^r(t)\big)}_{\triangleq\, \widehat{x}_j(t+1)} - a_{ij}\widehat{s}_i(t)\,\underbrace{\mu_j^r(t) g'_{\mathsf{in}}\big(\widehat{r}_j(t), q_j, \mu_j^r(t)\big)}_{\triangleq\, \mu_j^x(t+1)}, \tag{2.62}$$

where a first-order Taylor-series approximation was used in (2.62). We have neglected $O(1/N)$ terms and note that the remaining correction to $\widehat{x}_j(t+1)$ is $O(1/\sqrt{N})$, as expected from comparison to the derivations in [6,16,17]. It should be noted that the definitions of $\widehat{x}_j(t+1)$ and $\mu_j^x(t+1)$ below (2.62) are consistent with those of $\widehat{x}_{ij}(t+1)$ and $\mu_{ij}^x(t+1)$ in (2.52) and (2.53), respectively.

Finally, the updates to $\widehat{p}_i(t)$ and $\mu_i^p(t)$ (recall (2.16) and (2.17)) can be approximated as follows:

$$\widehat{p}_i(t+1) \;\triangleq\; \sum_j a_{ij}\widehat{x}_{ij}(t+1) \;\approx\; \sum_j a_{ij}\widehat{x}_j(t+1) - \widehat{s}_i(t)\underbrace{\sum_j a_{ij}^2\mu_j^x(t+1)}_{\approx\,\mu_i^p(t+1)}. \quad (2.63)$$

### 2.4.3 Summary of MMSE GAMP

Definitions:

$$p_{Z|Y,P}(z_i \,|\, y_i, \widehat{p}_i; \mu_i^p) \;\propto\; p_{Y|Z}(y_i|z_i)\mathcal{N}(z_i; \widehat{p}_i, \mu_i^p) \tag{2.64}$$

$$p_{X|Q,R}(x_j|q_j, \widehat{r}_j; \mu_j^r) \;\propto\; p_{X|Q}(x_j|q_j)\mathcal{N}(x_j; \widehat{r}_j, \mu_j^r) \tag{2.65}$$

Initialization ($t = 1$):

$$\forall j: \quad \widehat{x}_j(1) \;=\; \int x\, p_{X|Q}(x|q_j)dx \tag{2.66}$$

$$\forall j: \quad \mu_j^x(1) \;=\; \int |x - \widehat{x}_j(1)|^2\, p_{X|Q}(x|q_j)dx \tag{2.67}$$

$$\forall i: \quad \widehat{s}_i(0) \;=\; 0 \tag{2.68}$$

Output nodes ($t \geq 1$):

$$\forall i: \quad \mu_i^p(t) \;=\; \textstyle\sum_{j=1}^n |a_{ij}|^2 \mu_j^x(t) \tag{2.69}$$

$$\forall i: \quad \widehat{p}_i(t) \;=\; \sum_{j=1}^n a_{ij}\widehat{x}_j(t) - \mu_i^p(t)\widehat{s}_i(t-1) \tag{2.70}$$

$$\forall i: \quad \widehat{z}_i(t) \;=\; \mathrm{E}\{z_i \,|\, y_i, \widehat{p}_i(t); \mu_i^p(t)\} \tag{2.71}$$

$$\forall i: \quad \mu_i^z(t) \;=\; \mathrm{var}\{z_i \,|\, y_i, \widehat{p}_i(t); \mu_i^p(t)\} \tag{2.72}$$

$$\forall i: \quad \widehat{s}_i(t) \;=\; \big(\widehat{z}_i(t) - \widehat{p}_i(t)\big)/\mu_i^p(t) \tag{2.73}$$

$$\forall i: \quad \mu_i^s(t) \;=\; \big(1 - \mu_i^z(t)/\mu_i^p(t)\big)/\mu_i^p(t) \tag{2.74}$$

Input nodes ($t \geq 1$):

$$\forall j: \quad \mu_j^r(t) \;=\; \Big(\textstyle\sum_{i=1}^m |a_{ij}|^2 \mu_i^s(t)\Big)^{-1} \tag{2.75}$$

$$\forall j: \quad \widehat{r}_j(t) \;=\; \widehat{x}_j(t) + \mu_j^r(t) \textstyle\sum_{i=1}^m a_{ij}^* \widehat{s}_i(t) \tag{2.76}$$

$$\forall j: \quad \widehat{x}_j(t{+}1) \;=\; \mathrm{E}\{x_j \,|\, q_j, \widehat{r}_j(t); \mu_j^r(t))\} \tag{2.77}$$

$$\forall j: \quad \mu_j^x(t{+}1) \;=\; \mathrm{var}\{x_j \,|\, q_j, \widehat{r}_j(t); \mu_j^r(t))\} \tag{2.78}$$

The above algorithm is summarized using absolute values and conjugates to facilitate operation in the complex-valued case, in which case the definitions (2.64)-(2.65) would use circular complex Gaussian distributions. We also note that the rigorous GAMP analysis presented in [6] requires several small modifications of the above algorithm, some of which assume that the columns of $\boldsymbol{A}$ are approximately unit-norm.

## 2.5 Example Channel Models for MMSE GAMP

In this section we provide examples of the prior on the unknown signal (2.1), which Rangan refers to as an input channel, and the likelihood function (2.2), which Rangan refers to as the output channel.

In later chapters, we will consider a variety of likelihood functions, including Gaussian mixtures and Laplacian noise. However, the most common choice for the output channel is Additive White Gaussian Noise (AWGN), i.e., $p_{Y|Z}(y \,|\, z) = \mathcal{N}(y; z, \mu^w)$ for noise variance $\mu^w$. The required mean and variance calculations to implement GAMP for this channel model are given as

$$\mathrm{E}\{z \,|\, y, \widehat{p}; \mu^p\} = \widehat{p} + \frac{\mu^p}{\mu^p + \mu^w}(y - \widehat{p}) \tag{2.79}$$

$$\mathrm{var}\{z \,|\, y, \widehat{p}; \mu^p\} = \frac{1}{1/\mu^p + 1/\mu^w}. \tag{2.80}$$

Note that the complex-valued circular-AWGN output channel yields the same expressions.

For the input channel, one of the most common choices will be the sparsity inducing Bernoulli-Gaussian distribution given by $p_{X|Q}(x \,|\, q = [\lambda, \widehat{\theta}, \mu^\theta]) = \lambda\mathcal{N}(x; \widehat{\theta}, \mu^\theta) + (1-\lambda)\delta(x)$, where $\widehat{\theta}, \mu^\theta$ are the mean and variance, respectively, of the non-zero components, and $\lambda \in [0, 1]$ controls the sparsity rate. The required mean and variance

expressions to implement this channel model are given by

$$\mathrm{E}\{x|q,\widehat{r};\mu^r\} = \frac{\gamma}{\alpha} \tag{2.81}$$

$$\mathrm{var}\{x|q,\widehat{r};\mu^r\} = \gamma^2\frac{\alpha-1}{\alpha^2}+\frac{\nu}{\alpha}, \tag{2.82}$$

for

$$\alpha = 1+\frac{1-\lambda}{\lambda}\sqrt{\frac{\mu^\theta}{\nu}}\exp\left(\frac{1}{2}\left[\frac{(\widehat{r}-\widehat{\theta})^2}{\mu^\theta+\mu^r}-\frac{\widehat{r}^2}{\mu^r}\right]\right) \tag{2.83}$$

$$\gamma = \frac{\widehat{\theta}/\mu^\theta+\widehat{r}/\mu^r}{1/\mu^\theta+1/\mu^r} \tag{2.84}$$

$$\nu = \frac{1}{1/\mu^r+1/\mu^\theta}. \tag{2.85}$$

Note that, when $\widehat{\theta}=0$, some equations can be simplified:

$$\alpha = 1+\frac{1-\lambda}{\lambda}\sqrt{\frac{\mu^\theta}{\nu}}\exp\left(-\frac{1}{2}\frac{\gamma^2}{\nu}\right) \tag{2.86}$$

$$\gamma = \frac{\nu\widehat{r}}{\mu^r}. \tag{2.87}$$

For Bernoulli-$\mathcal{CN}$, we remove the above square-roots and $\frac{1}{2}$ terms, and change squares to absolute-squares.

## 2.6 Damping for Numerical Robustness

GAMP was derived and analyzed for random, iid matrices $\boldsymbol{A}$. However, the use of "damping" with GAMP yields provable convergence guarantees with arbitrary matrices [18]. Practical implementations of GAMP thus use an adaptive step size procedure to ensure convergence. A step size $\beta$, marked in red for clarity, is inserted into the algorithm to obtain

Initialization ($t = 1$):

$$\forall j: \quad \widehat{x}_j(1) = \int x\, p_{X|Q}(x|q_j)dx \tag{2.88}$$

$$\forall j: \quad \mu_j^x(1) = \int |x - \widehat{x}_j(1)|^2\, p_{X|Q}(x|q_j)dx \tag{2.89}$$

$$\forall i: \quad \widehat{s}_i(0) = 0 \tag{2.90}$$

Output nodes ($t \geq 1$):

$$\mu_i^p(t) = \beta \textstyle\sum_{j=1}^n |a_{ij}|^2 \mu_j^x(t) + (1-\beta)\mu_i^p(t-1) \tag{2.91}$$

$$\widehat{p}_i(t) = \sum_{j=1}^n a_{ij}\widehat{x}_j(t) - \mu_i^p(t)\widehat{s}_i(t-1) \tag{2.92}$$

$$\widehat{s}_i(t) = \beta g_{\mathsf{out}}\big(\widehat{p}_i(t), y_i, \mu_i^p(t)\big) + (1-\beta)\widehat{s}_i(t-1) \tag{2.93}$$

$$\mu_i^s(t) = -\beta g'_{\mathsf{out}}\big(\widehat{p}_i(t), y_i, \mu_i^p(t)\big) + (1-\beta)\mu_i^s(t-1) \tag{2.94}$$

Input nodes ($t \geq 1$):

$$\overline{x}_j(t) = \beta\widehat{x}_j(t) + (1-\beta)\overline{x}_j(t-1) \tag{2.95}$$

$$\mu_j^r(t) = \Big(\textstyle\sum_{i=1}^m |a_{ij}|^2 \mu_i^s(t)\Big)^{-1} \tag{2.96}$$

$$\widehat{r}_j(t) = \overline{x}_j(t) + \mu_j^r(t) \textstyle\sum_{i=1}^m a_{ij}^* \widehat{s}_i(t) \tag{2.97}$$

$$\widehat{x}_j(t+1) = g_{\mathsf{in}}(\widehat{r}_j(t), q_j, \mu_j^r(t)) \tag{2.98}$$

$$\mu_j^x(t+1) = \mu_j^r(t)\, g'_{\mathsf{in}}(\widehat{r}_j(t), q_j, \mu_j^r(t)), \tag{2.99}$$

where we have written the algorithm in terms of the previously defined $g_{\mathsf{in}}$ function. We also use a similar $g_{\mathsf{out}}$ notation to represent the output channel. In particular, $g_{\mathsf{out}}\big(\widehat{p}_i(t), y_i, \mu_i^p(t)\big) \triangleq H'\big(\widehat{p}_i(t), y_i, \mu_i^p(t)\big)$, which, recalling (2.21) and (2.22), implies that $-g'_{\mathsf{out}}\big(\widehat{p}_i(t), y_i, \mu_i^p(t)\big) = \big(1 - \mu_i^z(t)/\mu_i^p(t)\big)/\mu_i^p(t)$.

Notice that the algorithm now contains an extra state, namely $\overline{x}_j(t)$. To avoid choosing an initialization for this state, $\beta = 1$ for the first time step. In general,

some schemes allow $\beta$ to vary with the time step, while others use a constant step, with the exception of $t = 1$. The step is taken in all of the quantities that are used to compute $\widehat{r}_j(t)$. In this way, a step size of zero results in no change in the current state. Trivially, we note that $\beta = 1$ for all time steps reduces to the standard GAMP algorithm. Finally, the practical implementation uses a step-acceptance procedure to verify that a given step improves the cost function described in Section 2.8 before updating the actual state variables.

## 2.7 Additional Modifications for Numerical Robustness

Particularly for problems at very high signal to noise ratios, several of the variances tracked by the GAMP algorithm take on extreme values over a potentially problematic dynamic range. In particular, $\boldsymbol{\mu}^s$ becomes very large, while $\boldsymbol{\mu}^x$, $\boldsymbol{\mu}^r$, and $\boldsymbol{\mu}^p$ become extremely small. Our practical implementation bounds the small variances from below, but it may be beneficial to track scaled versions of these variances to avoid potential numerical problems. Here we develop a slightly modified algorithm to accomplish this goal.

Initialization ($t = 1$):

$$\forall j: \quad \widehat{x}_j(1) = \int x\, p_{X|Q}(x|q_j)dx \tag{2.100}$$

$$\forall j: \quad \mu_j^x(1) = \int |x - \widehat{x}_j(1)|^2\, p_{X|Q}(x|q_j)dx \tag{2.101}$$

$$\forall i: \quad \widehat{s}_i(0) = 0 \tag{2.102}$$

Output nodes ($t \geq 1$):

$$\mu_i^p(t) = \beta \sum_{j=1}^n |a_{ij}|^2 \mu_j^x(t) + (1-\beta)\mu_i^p(t-1) \tag{2.103}$$

$$< \mu^p(t) > = \frac{1}{m}\sum_{i=1}^m \mu_i^p(t) \tag{2.104}$$

$$\widehat{p}_i(t) = \sum_{j=1}^n a_{ij}\widehat{x}_j(t) - \frac{\mu_i^p(t)}{< \mu^p(t-1) >}\widehat{\underline{s}}_i(t-1) \tag{2.105}$$

$$\widehat{\underline{s}}_i(t) = \beta < \mu^p(t) > g_{\mathsf{out}}\big(\widehat{p}_i(t), y_i, \mu_i^p(t)\big) + (1-\beta)\widehat{\underline{s}}_i(t-1) \tag{2.106}$$

$$\underline{\mu}_i^s(t) = -\beta < \mu^p(t) > g'_{\mathsf{out}}\big(\widehat{p}_i(t), y_i, \mu_i^p(t)\big) + (1-\beta)\underline{\mu}_i^s(t-1) \tag{2.107}$$

Input nodes ($t \geq 1$):

$$\overline{x}_j(t) = \beta\widehat{x}_j(t) + (1-\beta)\overline{x}_j(t-1) \tag{2.108}$$

$$\underline{\mu}_j^r(t) = \Big(\sum_{i=1}^m |a_{ij}|^2 \underline{\mu}_i^s(t)\Big)^{-1} \tag{2.109}$$

$$\widehat{r}_j(t) = \overline{x}_j(t) + \underline{\mu}_j^r(t)\sum_{i=1}^m a_{ij}^* \widehat{\underline{s}}_i(t) \tag{2.110}$$

$$\widehat{x}_j(t+1) = g_{\mathsf{in}}(\widehat{r}_j(t), q_j, < \mu^p(t) > \underline{\mu}_j^r(t)) \tag{2.111}$$

$$\mu_j^x(t+1) = < \mu^p(t) > \underline{\mu}_j^r(t)\, g'_{\mathsf{in}}(\widehat{r}_j(t), q_j, < \mu^p(t) > \underline{\mu}_j^r(t)) \tag{2.112}$$

Notice that we are now computing normalized versions of $\widehat{\underline{s}}_i(t)$, $\underline{\mu}_i^s(t)$, and $\underline{\mu}_j^r(t)$. If we fix $\beta = 1$ for all iterations, then the normalizations by $< \mu^p(t) >$ have no effect on the algorithm analytically. Notice in particular that the scale factors in the calculation of $\widehat{r}_j(t)$ cancel. We have also added a step size into the calculation of $\mu_i^p(t)$.

## 2.8 Cost Function for MMSE-GAMP

In Section 2.6, we described a method for improving GAMP's convergence behavior using an adaptive step scheme. In this section, we describe the cost function

used in the GAMPmatlab [19] implementation of MMSE GAMP to test whether a candidate step will be accepted. In particular, a step is "accepted" if it reduces the value of the cost function[3].

The first goal is to formulate the cost function associated with MMSE GAMP, i.e., an optimization problem whose critical points are the fixed points of MMSE GAMP. Following Rangan's development from [20], we first notice that (trivially)

$$p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{y}, \boldsymbol{q}) = \underset{b(\boldsymbol{x},\boldsymbol{z})}{\arg\min}\, D(b(\boldsymbol{x}, \boldsymbol{z}) \,\|\, p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{y}, \boldsymbol{q})) \tag{2.113}$$

where $D(p_1\|p_2)$ denotes the KL divergence between distributions $p_1$ and $p_2$. Furthermore, because the joint posterior enforces the constraint $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{x}$ via $p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{y}, \boldsymbol{q}) = C(\boldsymbol{y})p(\boldsymbol{x}|\boldsymbol{q})p(\boldsymbol{y}|\boldsymbol{z})\delta(\boldsymbol{z} - \boldsymbol{A}\boldsymbol{x})$, we can safely restrict the distributions $b(\boldsymbol{x}, \boldsymbol{z})$ in the minimization above to the form $b(\boldsymbol{x}, \boldsymbol{z}) = b^x(\boldsymbol{x})\delta(\boldsymbol{z} - \boldsymbol{A}\boldsymbol{x})$, in which case

$$D(b(\boldsymbol{x}, \boldsymbol{z}) \,\|\, p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{y}, \boldsymbol{q})) = \int_{\boldsymbol{x},\boldsymbol{z}} b(\boldsymbol{x}, \boldsymbol{z}) \log \frac{b(\boldsymbol{x}, \boldsymbol{z})}{p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{y}, \boldsymbol{q})} \tag{2.114}$$

$$= \int_{\boldsymbol{x},\boldsymbol{z}} b(\boldsymbol{x}, \boldsymbol{z}) \log \frac{b^x(\boldsymbol{x})\delta(\boldsymbol{z} - \boldsymbol{A}\boldsymbol{x})}{C(\boldsymbol{y})p(\boldsymbol{x}|\boldsymbol{q})p(\boldsymbol{y}|\boldsymbol{z})\delta(\boldsymbol{z} - \boldsymbol{A}\boldsymbol{x})} \tag{2.115}$$

$$= \int_{\boldsymbol{x},\boldsymbol{z}} b(\boldsymbol{x}, \boldsymbol{z}) \log \frac{b^x(\boldsymbol{x})}{C(\boldsymbol{y})p(\boldsymbol{x}|\boldsymbol{q})p(\boldsymbol{y}|\boldsymbol{z})} \tag{2.116}$$

$$= \int_{\boldsymbol{x}} b^x(\boldsymbol{x}) \log \frac{b^x(\boldsymbol{x})}{p(\boldsymbol{x}|\boldsymbol{q})} + \int_{\boldsymbol{z}} b^z(\boldsymbol{z}) \log \frac{1}{p(\boldsymbol{y}|\boldsymbol{z})} - \log C(\boldsymbol{y}) \tag{2.117}$$

$$= \int_{\boldsymbol{x}} b^x(\boldsymbol{x}) \log \frac{b^x(\boldsymbol{x})}{p(\boldsymbol{x}|\boldsymbol{q})} + \int_{\boldsymbol{z}} b^z(\boldsymbol{z}) \log \frac{b^z(\boldsymbol{z})}{p(\boldsymbol{y}|\boldsymbol{z})Z(\boldsymbol{y})^{-1}} + \int_{\boldsymbol{z}} b^z(\boldsymbol{z}) \log \frac{1}{b^z(\boldsymbol{z})} - \log \frac{C(\boldsymbol{y})}{Z(\boldsymbol{y})^{-1}} \tag{2.118}$$

$$= D(b^x(\boldsymbol{x})\|p(\boldsymbol{x}|\boldsymbol{q})) + D(b^z(\boldsymbol{z})\|p(\boldsymbol{y}|\boldsymbol{z})) + H(b^z) + \mathsf{const} \tag{2.119}$$

where $H(p_1)$ is the entropy of distribution $p_1$ and $Z(\boldsymbol{y}) = \int_{\boldsymbol{z}} p(\boldsymbol{y} \,|\, \boldsymbol{z})$ is a scaling constant. Then, because the objective function in (2.113) has decoupled into terms

[3]An enhancement allows the cost function to increase over a small window of iterations.

that involve only $b^x(\boldsymbol{x})$ and $b^z(\boldsymbol{z})$, the optimization problem (2.113) can be restated as

$$(p(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{q}), p(\boldsymbol{z} \mid \boldsymbol{y}, \boldsymbol{q})) = \underset{b^x, b^z}{\arg\min}\, J_{KL}(b^x, b^z) \qquad \text{s.t.} \qquad b^z = T_{\boldsymbol{A}} b^x, \tag{2.120}$$

$$J_{KL}(b^x, b^z) = D\bigg(b^x \,\bigg\|\, \prod_{j=1}^{N} p_{X|Q}(\cdot|q_j)\bigg) + D\bigg(b^z \,\bigg\|\, \prod_{i=1}^{M} p_{Y|Z}(y_i \,|\, \cdot) Z_i^{-1}\bigg) + H\big(b^z\big), \tag{2.121}$$

where the constraint $b^z = T_{\boldsymbol{A}} b^x$ indicates that $b^z$ is the density of $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{x}$ where $x \sim \boldsymbol{b}^x$.

Unfortunately, the *approximate* marginal posteriors computed by GAMP are not critical points of (2.120). Rangan thus developed an approximation to (2.120) such that MMSE GAMP converges to a fixed point of the modified problem. The approximation involves three steps. First, separable approximations to the true marginal posteriors are employed. Thus, the optimization is restricted to separable densities[4] of the form

$$b^x(\boldsymbol{x}) = \prod_{j=1}^{N} b_j^x(x_j) \tag{2.122}$$

$$b^z(\boldsymbol{z}) = \prod_{i=1}^{M} b_i^z(z_i). \tag{2.123}$$

Second, the entropy $H(b_i^z)$ is replaced with a Gaussian upper bound $H_G(b_i^z, \mu_i^p)$ to yield the modified objective function

$$J_{SP}(b^x, b^z, \boldsymbol{\mu^p}) = D\bigg(b^x \,\bigg\|\, \prod_{j=1}^{N} p_{X|Q}(\cdot|q_j)\bigg) + D\bigg(b^z \,\bigg\|\, \prod_{i=1}^{M} p_{Y|Z}(y_i \,|\, \cdot) Z_i^{-1}\bigg) + \sum_{i=1}^{M} H_G\big(b_i^z, \mu_i^p\big) \tag{2.124}$$

$$H_G\big(b_i^z, \mu_i^p\big) \triangleq \frac{\operatorname{var}(z_i \,|\, z_i \sim b_i^z)}{2\mu_i^p} + \frac{\log 2\pi\mu_i^p}{2}. \tag{2.125}$$

[4]This approximation is analogous to the *mean field* approximation used in physics.

Using the entropy maximizing property of the Gaussian distribution, it is straightforward to prove that $H(b_i^z) \leq H_G\big(b_i^z, \mu_i^p\big)$, with equality when $b_i^z$ is Gaussian with variance $\mu_i^p$. As a result, for any positive vector $\boldsymbol{\mu}^p$ and densities[5] $b^x$ and $b^z$, we have the upper bound

$$J_{SP}(b^x, b^z, \boldsymbol{\mu}^p) \geq J_{KL}(b^x, b^z). \tag{2.126}$$

The third approximation involves weakening the constraint $b^z = T_{\boldsymbol{A}} b^x$. The separable approximate marginal distributions will in general not satisfy this constraint. Instead, the constraint is replaced with a weaker pair of moment matching constraints: $\mathrm{E}\{\boldsymbol{z} \,|\, b^z\} = \mathrm{E}\{\boldsymbol{A}\boldsymbol{x} \,|\, b^x\} = \boldsymbol{A}\,\mathrm{E}\{\boldsymbol{x} \,|\, b^x\}$ and $\boldsymbol{\mu}^p = \mathrm{var}\{\boldsymbol{A}\boldsymbol{x} \,|\, b^x\} = \boldsymbol{A}^2\,\mathrm{var}\{\boldsymbol{x} \,|\, b^x\}$. Here, the square on $\boldsymbol{A}^2$ is understood to be taken component-wise, and the $\mathrm{var}\{\cdot\}$ represents the vector of component variances, rather than a covariance matrix.

Combining all three of these approximations, Rangan obtains the optimization problem

$$\begin{aligned}\left(\widehat{b}^x, \widehat{b}^z, \boldsymbol{\mu}^p\right) = \underset{b^x, b^z, \overline{\boldsymbol{\mu}}^p}{\arg\min}\, & J_{SP}(b^x, b^z, \overline{\boldsymbol{\mu}}^p) \\ \text{s.t.} \quad & \mathrm{E}\{\boldsymbol{z} \,|\, b^z\} = \boldsymbol{A}\,\mathrm{E}\{\boldsymbol{x} \,|\, b^x\}, \quad \boldsymbol{\mu^p} = \boldsymbol{A}^2\,\mathrm{var}\{\boldsymbol{x} \,|\, b^x\}.\end{aligned} \tag{2.127}$$

As Rangan points out, the resulting minimum is neither an upper or lower bound to that of (2.120); while $J_{SP}$ is an upper bound to $J_{KL}$, the constraints have been weakened. However, Rangan proves a claim that if MMSE GAMP converges, then

[5]The bound holds even if the densities are not separable.

its approximated marginal posteriors

$$\widehat{p}(\boldsymbol{x}|\boldsymbol{y}) = \prod_{j=1}^{N} p_{X|Q,R}(x_j \mid q_j, \widehat{r}_j; \mu_j^r) = \prod_{j=1}^{N} p_{X|Q}(x_j \mid q_j)\mathcal{N}(x_j; \widehat{r}_j, \mu_j^r)C_j^{-1} \tag{2.128}$$

$$\text{for } C_j \triangleq \int_{x_j} p_{X|Q}(x_j \mid q_j)\mathcal{N}(x_j; \widehat{r}_j, \mu_j^r)$$

$$\widehat{p}(\boldsymbol{z}|\boldsymbol{y}) = \prod_{i=1}^{M} p_{Z|Y,P}(z_i \mid y_i, \widehat{p}_i; \mu_i^p) = \prod_{i=1}^{M} p_{Y|Z}(y_i \mid z_i)\mathcal{N}(z_i; \widehat{p}_i, \mu_i^p)B_i^{-1} \tag{2.129}$$

$$\text{for } B_i \triangleq \int_{z_i} p_{Y|Z}(y_i \mid z_i)\mathcal{N}(z_i; \widehat{p}_i, \mu_i^p)$$

along with its variances $\boldsymbol{\mu}^p$ are critical points of the problem (2.127). This correspondence motivates us to consider $J_{SP}$ as a cost function for step size selection in MMSE GAMP.

At this point, we pause to make a few comments about the relationships between some of GAMP's signal estimates. We define the mean and component-wise variance of (2.129) as $\widehat{\boldsymbol{z}}(t)$ and $\boldsymbol{\mu}^z(t)$, respectively. Notice that these are the mean and variance quantities that are used to compute $g_{\mathsf{out}}$ and $g'_{\mathsf{out}}$. Similarly, we will refer to the mean and variance of (2.128), which are the quantities computed by $g_{\mathsf{in}}$ and $g'_{\mathsf{in}}$, as $\widehat{\boldsymbol{x}}(t)$ and $\boldsymbol{\mu}^x(t)$. A natural question arises as to how these estimates are related.

We have just seen that GAMP provides a solution to (2.127) when it converges. Thus, from the first moment matching constraint, we know that $\boldsymbol{A}\widehat{\boldsymbol{x}}(t) = \widehat{\boldsymbol{z}}(t)$ *at convergence*. This equivalence can be easily seen by writing out the update equation for $\widehat{\boldsymbol{s}}(t)$ from (2.73) in terms of these estimates,

$$\widehat{\boldsymbol{s}}(t) = [\widehat{\boldsymbol{z}}(t) - \boldsymbol{A}\widehat{\boldsymbol{x}}(t) + \boldsymbol{\mu}^p(t) \odot \widehat{\boldsymbol{s}}(t-1)] \oslash \boldsymbol{\mu}^p(t), \tag{2.130}$$

where $\odot$ and $\oslash$ indicate element-wise multiplication and division, respectively. Clearly, $\widehat{\boldsymbol{s}}(t) = \widehat{\boldsymbol{s}}(t-1)$ once the two estimates are equal[6]. Indeed, one might consider using $\|\widehat{\boldsymbol{z}}(t) - \boldsymbol{A}\widehat{\boldsymbol{x}}(t)\|_F^2$ as a convergence criterion for GAMP. In contrast, we emphasize that in general $\widehat{\boldsymbol{p}}(t) \neq \widehat{\boldsymbol{z}}(t)$, even at convergence, since $\widehat{\boldsymbol{p}}(t)$ includes an additional Onsager correction term, i.e., the $\mu_i^p(t)\widehat{s}_i(t-1)$ in (2.70), based on $\widehat{\boldsymbol{s}}(t-1)$. This relationship between $\widehat{\boldsymbol{z}}(t)$ and $\widehat{\boldsymbol{p}}(t)$ is clear from the form of the approximate posterior (2.129).

Finally, we note that the $\boldsymbol{\mu}^p(t) = \boldsymbol{A}^2\boldsymbol{\mu}^x(t)$ variances are distinct from, and larger than, the $\boldsymbol{\mu}^z(t)$ variances. As a simple example, $\boldsymbol{\mu}^z(t)$ will be zero for all $t$ in the case of noise-free measurements, whereas $\boldsymbol{\mu}^p(t)$ will only approach zero as the algorithm converges in this noiseless case.

With these comments in mind, we are finally prepared to state the cost function for MMSE GAMP step size selection. Rangan bases the cost function on $J_{SP}$. Rangan's implementation evaluates the cost function at the approximated posterior (2.128), but he elects to use a Gaussian approximation in place of the approximate posterior (2.129). In particular, he considers the Gaussian approximated posterior given by

$$\widehat{p}(\boldsymbol{z} \,|\, \boldsymbol{y}) \approx \prod_{i=1}^{M} p_{Z|\overline{P}}(z_i \,|\, \overline{p}_i; \mu_i^p) = \prod_{i=1}^{M} \mathcal{N}(z_i; \overline{p}_i, \mu_i^p) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{A}\widehat{\boldsymbol{x}}, \operatorname{diag}(\boldsymbol{A}^2\boldsymbol{\mu}^x)), \quad (2.131)$$

where we use the definition $\overline{\boldsymbol{p}}(t) \triangleq \boldsymbol{A}\widehat{\boldsymbol{x}}(t)$. This choice for the approximate posterior on $\boldsymbol{z}$ satisfies the first moment matching constraint in (2.127) for all $t$. In addition, for this Gaussian approximation, the bound (2.126) is tight. Put another way, we have $H(b_i^z) = H_G\big(b_i^z, \mu_i^p\big)$. This choice also simplifies the implementation, since various step sizes can be tested without recomputing the $g_{\mathsf{out}}$ functions. Notice also from

[6]The cancelation of $\boldsymbol{\mu}^p(t)$ in this expression suggests why allowing these variances to become arbitrarily small can lead to numerical problems.

the last equality in (2.131) that this approximate posterior for $\boldsymbol{z}$ can be written as a simple function of the moments of (2.128). Finally, the simple Gaussian functional form simplifies the evaluation of $J_{SP}$.

The reader may wonder why $\boldsymbol{\mu}^p(t)$ is used as the variance in this approximation, rather than $\boldsymbol{\mu}^z(t)$. There are at least two motivations for this choice. First, the use of $\boldsymbol{\mu}^p(t)$ is necessary to make the bound (2.126) tight. Second, the $\boldsymbol{\mu}^z(t)$ variances may be very small even for small $t$, potentially leading to numerical difficulties. As already pointed out, in the noise free case, $\boldsymbol{\mu}^z(t)$ is zero for all time. Thus, $\boldsymbol{\mu}^z(t)$ may not capture the uncertainty in the $\widehat{\boldsymbol{x}}(t)$ estimate prior to convergence.

Plugging in (2.131) and (2.128) into the objective function from (2.127), we obtain Rangan's cost function for MMSE GAMP step size selection as

$$J = \sum_{j=1}^{N} D\big(p_{X|Q,R}(\cdot \,|\, q_j, \widehat{r}_j; \mu_j^r) \;\big\|\; p_{X|Q}(\cdot|q_j)\big) + \sum_{i=1}^{M} D\big(p_{Z|\overline{P}}(\cdot \,|\, \overline{p}_i; \mu_i^p) \;\big\|\; p_{Y|Z}(y_i \,|\, \cdot) Z_i^{-1}\big) + \sum_{i=1}^{m} H\big(p_{Z|\overline{P}}(\cdot \,|\, \overline{p}_i; \mu_i^p)\big) \tag{2.132}$$

$$= \sum_{j=1}^{N} D\big(p_{X|Q,R}(\cdot \,|\, q_j, \widehat{r}_j; \mu_j^r) \;\big\|\; p_{X|Q}(\cdot|q_j)\big) + \sum_{i=1}^{M} \left( \int_{z_i} p_{Z|\overline{P}}(z_i \,|\, \overline{p}_i; \mu_i^p) \log \frac{p_{Z|\overline{P}}(z_i \,|\, \overline{p}_i; \mu_i^p)_i}{p_{Y|Z}(y_i \,|\, z_i) Z_i^{-1}} - \int_{z_i} p_{Z|\overline{P}}(z_i \,|\, \overline{p}_i; \mu_i^p) \log p_{Z|\overline{P}}(z_i \,|\, \overline{p}_i; \mu_i^p) \right) \tag{2.133}$$

$$= \sum_{j=1}^{N} D\big(p_{X|Q,R}(\cdot|q_j, \widehat{r}_j; \mu_j^r) \;\big\|\; p_{X|Q}(\cdot|q_j)\big) - \mathrm{E}_{p_{Z|\overline{P}}(\cdot|\overline{p}_i;\mu_i^p)} \big\{ \log p_{Y|Z}(y_i \,|\, Z) \big\} + \text{const.} \tag{2.134}$$

The constant term depends only on the prior distributions and can be ignored by MMSE-GAMP. Notice that the second term in (2.134) is the (negative) expected log-likelihood. Next we show how the terms in (2.134) can be simplified.

### 2.8.1 Expected Log-likelihood of the measured data

The first term in (2.134) can then be simplified as follows:

$$\sum_{i=1}^{M} \mathrm{E}_{p_{Z|\overline{P}}(\cdot|\overline{p}_i;\mu_i^p)} \left\{ \log p_{Y|Z}(y_i \,|\, Z) \right\} = \sum_{i=1}^{M} \int_{z_i} \mathcal{N}(z_i; \overline{p}_i, \mu_i^p) \log p_{Y|Z}(y_i \,|\, z_i). \tag{2.135}$$

In the AWGN case, i.e., $p_{Y|Z}(y_i \,|\, z_i) = \mathcal{N}(y_i; z_i, \mu^w)$, we can simplify further:

$$\begin{aligned}
&\sum_{i=1}^{M} \mathrm{E}_{p_{Z|\overline{P}}(\cdot|\overline{p}_i;\mu_i^p)} \left\{ \log p_{Y|Z}(y_i \,|\, Z) \right\} \\
&= \sum_{i=1}^{M} \int_{z_i} \mathcal{N}(z_i; \overline{p}_i, \mu_i^p) \log \mathcal{N}(y_i; z_i, \mu^w) && (2.136) \\
&= -\sum_{i=1}^{M} \Big( \log \sqrt{2\pi\mu^w} + \frac{1}{2\mu^w} \int_{z_i} \mathcal{N}(z_i; \overline{p}_i, \mu_i^p) \, (z_i - y_i)^2 \Big). && (2.137)
\end{aligned}$$

Finally, using

$$\begin{aligned}
&\int_{z_i} \mathcal{N}(z_i; \overline{p}_i, \mu_i^p) \, (z_i - y_i)^2 \\
&= \int_{z_i} \mathcal{N}(z_i; \overline{p}_i, \mu_i^p) \, (z_i - \overline{p}_i + \overline{p}_i - y_i)^2 && (2.138) \\
&= \int_{z_i} \mathcal{N}(z_i; \overline{p}_i, \mu_i^p) \big( (z_i - \overline{p}_i)^2 + 2(z_i - \overline{p}_i)(\overline{p}_i - y_i) + (\overline{p}_i - y_i)^2 \big) \\
&= \mu_i^p + (\overline{p}_i - y_i)^2, && (2.139)
\end{aligned}$$

we get

$$\sum_{i=1}^{M} \mathrm{E}_{p_{Z|\overline{P}}(\cdot|\overline{p}_i;\mu_i^p)} \left\{ \log p_{Y|Z}(y_i \,|\, Z) \right\} = -\sum_{i=1}^{M} \Big( \log \sqrt{2\pi\mu^w} + \frac{1}{2\mu^w} \big( \mu_i^p + (\overline{p}_i - y_i)^2 \big) \Big). \tag{2.140}$$

Notice that the $\log \sqrt{2\pi\mu^w}$ can be neglected when $\mu^w$ is known, as GAMPmatlab does.

### 2.8.2 KL divergence from the prior

We now simplify the second term in (2.134):

$$
\begin{aligned}
&D\big(p_{X|Q,R}(\cdot|q_j,\widehat{r}_j;\mu_j^r)\;\big\|\;p_{X|Q}(\cdot|q_j)\big)\\
&=\int_{x_j} p_{X|Q,R}(x_j\,|\,q_j,\widehat{r}_j;\mu_j^r)\log\frac{p_{X|Q}(x_j\,|\,q_j)\mathcal{N}(x_j;\widehat{r}_j,\mu_j^r)C_j^{-1}}{p_{X|Q}(x_j|q_j)} &(2.141)\\
&=\int_{x_j} p_{X|Q,R}(x_j\,|\,q_j,\widehat{r}_j;\mu_j^r)\log\mathcal{N}(x_j;\widehat{r}_j,\mu_j^r)-\log C_j &(2.142)\\
&=-\Big(\log C_j+\log\sqrt{2\pi\mu_j^r}+\frac{1}{2\mu_j^r}\int_{x_j} p_{X|Q,R}(x_j\,|\,q_j,\widehat{r}_j;\mu_j^r)\,(x_j-\widehat{r}_j)^2\Big). &(2.143)
\end{aligned}
$$

Recalling that the pdf $p_{X|Q,R}(\cdot\,|\,q_j,\widehat{r}_j;\mu_j^r)$ has mean $\widehat{x}_j$ and variance $\mu_j^x$, we can write

$$
\begin{aligned}
&\int_{x_j} p_{X|Q,R}(x_j\,|\,q_j,\widehat{r}_j;\mu_j^r)\,(x_j-\widehat{r}_j)^2\\
&=\int_{x_j} p_{X|Q,R}(x_j\,|\,q_j,\widehat{r}_j;\mu_j^r)\,(x_j-\widehat{x}_j+\widehat{x}_j-\widehat{r}_j)^2 &(2.144)\\
&=\int_{x_j} p_{X|Q,R}(x_j\,|\,q_j,\widehat{r}_j;\mu_j^r)\Big((x_j-\widehat{x}_j)^2+2(x_j-\widehat{x}_j)(\widehat{x}_j-\widehat{r}_j)+(\widehat{x}_j-\widehat{r}_j)^2\Big) &(2.145)\\
&=\mu_j^x+(\widehat{x}_j-\widehat{r}_j)^2, &(2.146)
\end{aligned}
$$

so that

$$
\begin{aligned}
&\sum_{j=1}^{N} D\big(p_{X|Q,R}(\cdot|q_j,\widehat{r}_j;\mu_j^r)\;\big\|\;p_{X|Q}(\cdot|q_j)\big)\\
&\quad=-\sum_{j=1}^{N}\Big(\log C_j+\log\sqrt{2\pi\mu_j^r}+\frac{1}{2\mu_j^r}\big(\mu_j^x+(\widehat{x}_j-\widehat{r}_j)^2\big)\Big). &(2.147)
\end{aligned}
$$

A very simple form for this cost function is available when $p_{X|Q}(x_j\,|\,q_j)=\mathcal{N}(x_j;x_0,\mu_0^x)$. In this case, we notice that

$$
p_{X|Q,R}(x_j\,|\,q_j,\widehat{r}_j;\mu_j^r)=\mathcal{N}(x_j;\widehat{x}_j,\mu_j^x), \tag{2.148}
$$

which allows us to compute

$$\sum_{j=1}^{N} D\big(p_{X|Q,R}(\cdot|q_j,\widehat{r}_j;\mu_j^r) \;\big\|\; p_{X|Q}(\cdot|q_j)\big)$$

$$= \sum_{j=1}^{N} \int_{x_j} \mathcal{N}(x_j;\widehat{x}_j,\mu_j^x) \log \frac{\mathcal{N}(x_j;\widehat{x}_j,\mu_j^x)}{\mathcal{N}(x_j;x_0,\mu_0^x)} \tag{2.149}$$

$$= \sum_{j=1}^{N} \int_{x_j} \mathcal{N}(x_j;\widehat{x}_j,\mu_j^x) \left[\log\sqrt{\frac{\mu_0^x}{\mu_j^x}} - \frac{(x_j-\widehat{x}_j)^2}{2\mu_j^x} + \frac{(x_j-x_0)^2}{2\mu_0^x}\right] \tag{2.150}$$

$$= \sum_{j=1}^{N} \frac{1}{2}\left[\log\frac{\mu_0^x}{\mu_j^x} + \left(\frac{\mu_j^x}{\mu_0^x}-1\right) + \frac{(\widehat{x}_j-x_0)^2}{\mu_0^x}\right]. \tag{2.151}$$

A common situation arises when we wish to define a prior with a given sparsity level $\lambda$ and some arbitrary distribution for the non-zero entries. Rangan's `SparseScaEstim` class from GAMPmatlab specifically handles this case, defining the prior

$$p_{X|Q}(x|q) = (1-\lambda)\delta(x-\zeta) + \lambda f_X(x), \tag{2.152}$$

where the special case $\zeta = 0$ yields a sparsity inducing prior, and $f_X(x)$ is some arbitrary distribution, e.g., a Gaussian mixture. The advantage of the `SparseScaEstim` class is that it allows you to design an estimator based on the distribution $f_X(x)$ and then produce the corresponding estimator for (2.152) with no additional coding.

To see how this works, we first compute

$$C_j = \int_{x_j} p_{X|Q}(x_j \,|\, q_j)\mathcal{N}(x_j;\widehat{r}_j,\mu_j^r) \tag{2.153}$$

$$= (1-\lambda)\underbrace{\mathcal{N}(\zeta;\widehat{r}_j,\mu_j^r)}_{\triangleq C_j^0} + \lambda \underbrace{\int_{x_j} f_X(x)\mathcal{N}(x_j;\widehat{r}_j,\mu_j^r)}_{\triangleq C_j^1}. \tag{2.154}$$

The quantity $C_j^1$ is computed by the `plikey` method of an `EstimIn` object in the GAMPmatlab code. To understand why we refer to this quantity as a likelihood,

consider a binary indicator variable $\gamma_j$ that determines whether $x_j$ is on (drawn from the $f_X(x)$ distribution) or off ($x_j = \zeta$). With this hidden indicator variable defined, we see that $C_j^0$ and $C_j^1$ are precisely the likelihoods given $\widehat{r}_j$ that $\gamma_j$ is 0 or 1, respectively. Furthermore, we can define the posterior probabilities of $x_j$ being on or off given $\widehat{r}_j$ as

$$p(\gamma_j = 0 \,|\, \widehat{r}_j) = \frac{(1-\lambda)C_j^0}{C_j} \triangleq \pi_j^0 \tag{2.155}$$

$$p(\gamma_j = 1 \,|\, \widehat{r}_j) = \frac{\lambda C_j^1}{C_j} \triangleq \pi_j^1 = 1 - \pi_j^0. \tag{2.156}$$

If we are given an estimator designed for the prior $f_X(x)$, then it will compute the KL divergence

$$D^1(\widehat{\boldsymbol{r}}, \boldsymbol{\mu}^r) \triangleq D\left(\prod_{j=1}^N f_X(x_j)\mathcal{N}(x_j; \widehat{r}_j, \mu_j^r)(C_j^1)^{-1} \parallel \prod_{j=1}^N f_X(x_j)\right) \tag{2.157}$$

$$= \sum_{j=1}^N \underbrace{\int_{x_j} f_X(x_j)\mathcal{N}(x_j; \widehat{r}_j, \mu_j^r)(C_j^1)^{-1} \log\left(\frac{\mathcal{N}(x_j; \widehat{r}_j, \mu_j^r)}{C_j^1}\right)}_{\triangleq D_j^1(\widehat{r}_j, \mu_j^r)}, \tag{2.158}$$

where we leveraged the form (2.142) to obtain the second line. The estimator designed for the prior $f_X(x)$ will also return $C_j^1$ and estimates $\widehat{\boldsymbol{x}}^1$ and $\boldsymbol{\mu}^{(x,1)}$.

We also require the corresponding quantities for the off case, i.e., $\gamma_j = 0$ with the prior $\delta(x - \zeta)$. Fortunately, these quantities are readily computed as

$$D^0(\widehat{\boldsymbol{r}}, \boldsymbol{\mu}^r) \triangleq \int_{x_j} \delta(x_j - \zeta)\mathcal{N}(x_j; \widehat{r}_j, \mu_j^r)(C_j^0)^{-1} \log\left(\frac{\mathcal{N}(x_j; \widehat{r}_j, \mu_j^r)}{C_j^0}\right) = 0 \tag{2.159}$$

$$\widehat{x}_j^0 = \zeta \tag{2.160}$$

$$\mu_j^{x,0} = 0. \tag{2.161}$$

We can immediately obtain the approximate posterior moments as

$$\widehat{x}_j = \pi_j^0 \zeta + \pi_j^1 \widehat{x}_j^1 \tag{2.162}$$

$$\mu_j^x = \pi_j^0 \zeta^2 + \pi_j^1 \left[ (\widehat{x}_j^1)^2 + \mu_j^{(x,1)} \right] - \widehat{x}_j^2. \tag{2.163}$$

Our remaining goal is to find a simple expression for $\sum_{j=1}^N D\big(p_{X|Q,R}(\cdot|q_j, \widehat{r}_j; \mu_j^r) \;\big\|\; p_{X|Q}(\cdot|q_j)\big)$ in terms of the already available $D^1(\widehat{\boldsymbol{r}}, \boldsymbol{\mu}^r)$. Starting with the form (2.142), we obtain

$$\begin{aligned}
&\sum_{j=1}^N D\big(p_{X|Q,R}(\cdot|q_j, \widehat{r}_j; \mu_j^r) \;\big\|\; p_{X|Q}(\cdot|q_j)\big) \\
&= \sum_{j=1}^N \int_{x_j} p_{X|Q,R}(x_j \,|\, q_j, \widehat{r}_j; \mu_j^r) \log\left(\frac{\mathcal{N}(x_j; \widehat{r}_j, \mu_j^r)}{C_j}\right) && (2.164) \\
&= \sum_{j=1}^N \int_{x_j} (1-\lambda)\delta(x_j - \zeta)\mathcal{N}(x_j; \widehat{r}_j, \mu_j^r)(C_j)^{-1} \log\left(\frac{\mathcal{N}(x_j; \widehat{r}_j, \mu_j^r)}{C_j}\right) && (2.165) \\
&\quad + \lambda f_X(x_j)\mathcal{N}(x_j; \widehat{r}_j, \mu_j^r)(C_j)^{-1} \log\left(\frac{\mathcal{N}(x_j; \widehat{r}_j, \mu_j^r)}{C_j}\right) && (2.166) \\
&= \sum_{j=1}^N \pi_l^0 \int_{x_j} \delta(x_j - \zeta)\mathcal{N}(x_j; \widehat{r}_j, \mu_j^r)(C_j^0)^{-1} \left[\log\left(\frac{\mathcal{N}(x_j; \widehat{r}_j, \mu_j^r)}{C_j^0}\right) + \log\frac{C_j^0}{C_j}\right] && (2.167) \\
&\quad + \pi_l^1 \int_{x_j} f_X(x_j)\mathcal{N}(x_j; \widehat{r}_j, \mu_j^r)(C_j^1)^{-1} \left[\log\left(\frac{\mathcal{N}(x_j; \widehat{r}_j, \mu_j^r)}{C_j^1}\right) + \log\frac{C_j^1}{C_j}\right] && (2.168) \\
&= \sum_{j=1}^N \pi_j^0 \left[D_j^0(\widehat{r}_j, \mu_j^r) + \log\frac{\pi_j^0}{1-\lambda}\right] + \pi_j^1 \left[D_j^1(\widehat{r}_j, \mu_j^r) + \log\frac{\pi_j^1}{\lambda}\right] && (2.169) \\
&= \sum_{j=1}^N \pi_j^0 \log\frac{\pi_j^0}{1-\lambda} + \pi_j^1 \left[D_j^1(\widehat{r}_j, \mu_j^r) + \log\frac{\pi_j^1}{\lambda}\right]. && (2.170)
\end{aligned}$$

Thus we can compute the required KL divergence for the prior (2.152) given the output of an estimator designed for the prior $f_X(x)$. This completes the derivation of the procedure used by the `SparseScaEstim` class.

## 2.9 Connection to Donoho/Bayati/Montanari AMP

Rangan [6] states that his GAMP algorithm is equivalent to the AMP presented in [11] when considering the Gaussian output channel. Here we clarify this relationship between the algorithms. First, as pointed out already, under the AWGN or circular-AWGN output channel, we obtain

$$g_{\mathsf{out}}(\widehat{p}, y, \mu^p) = \frac{y - \widehat{p}}{\mu^p + \mu^w} \tag{2.171}$$

$$-g'_{\mathsf{out}}(\widehat{p}, y, \mu^p) = \frac{1}{\mu^p + \mu^w}. \tag{2.172}$$

GAMP is only equivalent to AMP given a few simplifications. In particular, we simplify the variance estimates to be the same for all nodes to obtain

$$\mu_i^p(t) = \sum_{j=1}^{N} |a_{ij}|^2 \mu_j^x(t) \approx \frac{1}{M} \sum_{j=1}^{N} \mu_j^x(t) \triangleq \mu^p(t) \tag{2.173}$$

$$\mu_i^s(t) = \frac{1}{\mu_i^p(t) + \mu^w} \approx \frac{1}{\mu^p(t) + \mu^w} \triangleq \mu^s(t) \tag{2.174}$$

$$\mu_j^r(t) = \left( \sum_{i=1}^{M} |a_{ij}|^2 \mu_i^s(t) \right)^{-1} \approx \frac{1}{\mu^s(t)} = \mu^p(t) + \mu^w \triangleq \mu^r(t). \tag{2.175}$$

These definitions yield simplified expressions for the $\mu_j^x$ variables as

$$\mu_j^x(t+1) = \mu_j^r(t)\, g'_{\mathsf{in}}(\widehat{r}_j(t), q_j, \mu_j^r(t)) \approx \mu^r(t)\, g'_{\mathsf{in}}(\widehat{r}_j(t), q_j, \mu^r(t)). \tag{2.176}$$

With these simplifications in mind, we can express GAMP in a more compact form. Notice that

$$\widehat{\boldsymbol{r}}(t) = \widehat{\boldsymbol{x}}(t) + \mu^r(t) \boldsymbol{A}^H \widehat{\boldsymbol{s}}(t) \tag{2.177}$$

$$= \widehat{\boldsymbol{x}}(t) + \mu^r(t) \boldsymbol{A}^H \left( \frac{\boldsymbol{y} - \widehat{\boldsymbol{p}}(t)}{\mu^p(t) + \mu^w} \right) \tag{2.178}$$

$$= \widehat{\boldsymbol{x}}(t) + \boldsymbol{A}^H \left( \boldsymbol{y} - \widehat{\boldsymbol{p}}(t) \right) \tag{2.179}$$

$$= \widehat{\boldsymbol{x}}(t) + \boldsymbol{A}^H \widehat{\boldsymbol{v}}(t), \tag{2.180}$$

where we have defined for convenience $\widehat{\boldsymbol{v}}(t) = \boldsymbol{y} - \widehat{\boldsymbol{p}}(t)$. We can express $\widehat{\boldsymbol{v}}(t)$ in a form that makes the AMP equivalence obvious. Consider

$$\begin{aligned}
\widehat{\boldsymbol{v}}(t) &= \boldsymbol{y} - \widehat{\boldsymbol{p}}(t) && (2.181)\\
&= \boldsymbol{y} - \boldsymbol{A}\widehat{\boldsymbol{x}}(t) - \mu^p(t)\frac{\boldsymbol{y} - \widehat{\boldsymbol{p}}(t-1)}{\mu^p(t-1) + \mu^w} && (2.182)\\
&= \boldsymbol{y} - \boldsymbol{A}\widehat{\boldsymbol{x}}(t) + \left(\frac{\mu^p(t)}{\mu^r(t-1)}\right)\widehat{\boldsymbol{v}}(t-1) && (2.183)\\
&= \boldsymbol{y} - \boldsymbol{A}\widehat{\boldsymbol{x}}(t) + \left(\frac{\frac{1}{M}\sum_{j=1}^{N}\mu_j^x(t)}{\mu^r(t-1)}\right)\widehat{\boldsymbol{v}}(t-1) && (2.184)\\
&= \boldsymbol{y} - \boldsymbol{A}\widehat{\boldsymbol{x}}(t) + \left(\frac{\frac{1}{M}\sum_{j=1}^{N}\mu^r(t-1)\, g'_{\mathsf{in}}(\widehat{r}_j(t-1), q_j, \mu^r(t-1))}{\mu^r(t-1)}\right)\widehat{\boldsymbol{v}}(t-1) && (2.185)\\
&= \boldsymbol{y} - \boldsymbol{A}\widehat{x}(t) + \frac{1}{\delta} < g'_{\mathsf{in}}(\widehat{\boldsymbol{r}}(t-1), \boldsymbol{q}, \mu^r(t-1)) > \widehat{\boldsymbol{v}}(t-1), && (2.186)
\end{aligned}$$

where $< \cdot >$ denotes the mean and $g_{\mathsf{in}}$ is understood to operate on vectors componentwise. As an aside, as suggested in [21], we can view $\widehat{\boldsymbol{v}}(t)$ as a filtered residual. Indeed, if we define the unfiltered residual $\widehat{\boldsymbol{b}}(t) = \boldsymbol{y} - \boldsymbol{A}\widehat{\boldsymbol{x}}(t)$, then we can explicitly compute $\widehat{\boldsymbol{v}}(t)$ as

$$\widehat{\boldsymbol{v}}(t) = \widehat{\boldsymbol{b}}(t) + \sum_{i=0}^{t-1}\left(\prod_{j=i+1}^{t}\frac{1}{\delta} < g'_{\mathsf{in}}(\widehat{\boldsymbol{r}}(j-1), \boldsymbol{q}, \mu^r(j-1)) >\right)\widehat{\boldsymbol{b}}(i). \qquad (2.187)$$

Thus, $\widehat{\boldsymbol{v}}(t)$ is the output of an IIR filter with time-varying coefficients acting on the unfiltered residuals.

Combining these results, we can express the complete scalar-variance AWGN-output GAMP (assuming $\mathrm{E}\{|A_{ij}|^2\} = \frac{1}{M}$) as

$$\widehat{\boldsymbol{v}}(t) = \boldsymbol{y} - \boldsymbol{A}\widehat{\boldsymbol{x}}(t) + \tfrac{1}{\delta}\tfrac{\mu^x(t)}{\mu^r(t-1)}\widehat{\boldsymbol{v}}(t-1) \qquad \text{Onsager-corrected residual} \qquad (2.188)$$

$$\widehat{\boldsymbol{r}}(t) = \widehat{\boldsymbol{x}}(t) + \boldsymbol{A}^H\widehat{\boldsymbol{v}}(t) \qquad \text{back-projection update} \qquad (2.189)$$

$$\mu^r(t) = \mu^w + \tfrac{1}{\delta}\mu^x(t) \qquad \text{error-variance of } \widehat{\boldsymbol{r}}(t) \qquad (2.190)$$

$$\widehat{\boldsymbol{x}}(t+1) = g_{\mathsf{in}}\big(\widehat{\boldsymbol{r}}(t), \boldsymbol{q}, \mu^r(t)\big) \qquad \text{nonlinear thresholding step} \qquad (2.191)$$

$$\mu^x(t+1) = \mu^r(t) < g'_{\mathsf{in}}\big(\widehat{\boldsymbol{r}}(t), \boldsymbol{q}, \mu^r(t)\big) > \qquad \text{error-variance of } \widehat{\boldsymbol{x}}(t+1) \qquad (2.192)$$

or more succinctly as

$$\widehat{\boldsymbol{v}}(t) = \boldsymbol{y} - \boldsymbol{A}\widehat{\boldsymbol{x}}(t) + \frac{1}{\delta} < g'_{\mathsf{in}}(\widehat{\boldsymbol{x}}(t-1) + \boldsymbol{A}^H\widehat{\boldsymbol{v}}(t-1), \boldsymbol{q}, \mu^r(t-1)) > \widehat{\boldsymbol{v}}(t-1) \qquad (2.193)$$

$$\widehat{\boldsymbol{x}}(t+1) = g_{\mathsf{in}}(\widehat{\boldsymbol{x}}(t) + \boldsymbol{A}^H\widehat{\boldsymbol{v}}(t), \boldsymbol{q}, \mu^r(t)). \qquad (2.194)$$

Comparing (2.193)-(2.194) to the DMM-AMP recursion presented in [3]:

$$\widehat{\boldsymbol{v}}(t) = \boldsymbol{y} - \boldsymbol{A}\widehat{\boldsymbol{x}}(t) + \frac{1}{\delta} < \eta'_{t-1}(\widehat{\boldsymbol{x}}(t-1) - \boldsymbol{A}^H\widehat{\boldsymbol{v}}(t-1)) > \widehat{\boldsymbol{v}}(t-1) \qquad (2.195)$$

$$\widehat{\boldsymbol{x}}(t+1) = \eta_t(\widehat{\boldsymbol{x}}(t) + \boldsymbol{A}^H\widehat{\boldsymbol{v}}(t)), \qquad (2.196)$$

we see that the recursions are equivalent when $\eta_t(\cdot) = g_{\mathsf{in}}(\cdot, \boldsymbol{q}, \mu^r(t))$.

That said, DMM have suggested various implementations of $\eta_t(\cdot)$, some of which are equivalent to what is done in the GAMP framework, and some of which are not. For example, the "Bayesian AMP" from DMM's [5, (23)-(25)] suggested (after fixing

typos) the recursion

$$\begin{aligned}
\widehat{\boldsymbol{v}}(t) &= \boldsymbol{y} - \boldsymbol{A}\widehat{\boldsymbol{x}}(t) \\
&\quad + \frac{1}{\delta} < g'_{\mathsf{in}}(\widehat{\boldsymbol{x}}(t-1) + \boldsymbol{A}^H\widehat{\boldsymbol{v}}(t-1), \boldsymbol{q}, \mu^w + \gamma(t-1)) > \widehat{\boldsymbol{v}}(t-1) \quad (2.197) \\
\widehat{\boldsymbol{x}}(t+1) &= g_{\mathsf{in}}\big(\widehat{\boldsymbol{x}}(t) + \boldsymbol{A}^H\widehat{\boldsymbol{v}}(t), \boldsymbol{q}, \mu^w + \gamma(t)\big) \quad (2.198) \\
\gamma(t+1) &= \frac{1}{\delta}(\mu^w + \gamma(t)) < g'_{\mathsf{in}}(\widehat{\boldsymbol{x}}(t) + \boldsymbol{A}^H\widehat{\boldsymbol{v}}(t), \boldsymbol{q}, \mu^w + \gamma(t)\big) > \quad (2.199)
\end{aligned}$$

which is identical to (2.193)-(2.194) via $\gamma(t) = \mu^x(t)/\delta$ and $\mu^w + \gamma(t) = \mu^r(t)$. On the other hand, for the LASSO problem, DMM suggested in [16, Sec. 9.5.1] using

$$\eta_t(\widehat{\boldsymbol{r}}(t)) = \eta(\widehat{\boldsymbol{r}}(t); \alpha\sqrt{\mu^v(t)}) \;\text{ for }\; \mu^v(t) = \frac{1}{M}\|\widehat{\boldsymbol{v}}(t)\|_2^2 \;\text{ or }\; \Big(\frac{1}{\Phi^{-1}(3/4)}|\widehat{\boldsymbol{v}}(t)|_{(m/2)}\Big)^2 \quad (2.200)$$

where $\eta(r, \theta)$ is the soft thresholding function with threshold value $\theta$, $|\widehat{\boldsymbol{v}}(t)|_{(m/2)}$ is the median magnitude, and $\Phi^{-1}(3/4) \approx 0.6745$ is the median[7] absolute value of a standard normal random variable. Here we note that i) the DMM threshold is set in proportion to a square-root variance (i.e., $\alpha\sqrt{\mu^v(t)}$), whereas GAMP uses a variance (i.e., $\lambda\mu^r(t)$), and ii) the DMM variance estimate $\mu^v(t)$ is computed directly from the residual $\widehat{\boldsymbol{v}}(t)$, whereas GAMP's variance estimate $\mu^r(t)$ is computed from $\eta'_{t-1}$ and $\mu^w$.

[7]DMM and their collaborators often prefer the median estimator. Note that this expression must be adjusted for complex-valued data, see [22].

# Chapter 3: Matrix Uncertain Generalized Approximate Message Passing

## 3.1 Introduction

As briefly described in Chapter 1, the goal in compressive sensing (CS) is to reconstruct an $N$-dimensional signal $\boldsymbol{x}$ from $M < N$ linear measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}$, where $\boldsymbol{w}$ is additive noise. In the noiseless case, it is by now well known that, when the signal is exactly $K$-sparse and the measurement matrix $\boldsymbol{A}$ satisfies certain properties (e.g., restricted isometry, null space, or spark), it is possible to exactly reconstruct the signal from $M = O(K \log N/K)$ measurements using polynomial-complexity algorithms (e.g., greedy or convex-optimization based). Moreover, these methods can accurately reconstruct the signal in the noisy case, even when the signal is compressible rather than exactly sparse (e.g., [23]).

These results are, however, predicated on knowing the measurement matrix $\boldsymbol{A}$ perfectly. In practical applications of CS, it is reasonable to expect uncertainty in the linear measurement matrix $\boldsymbol{A}$ due to, e.g., imperfections in the signal acquisition hardware, model mismatch, parameter discretization, and other factors.

Several authors have analyzed the impact of measurement-matrix uncertainty on existing CS algorithms, e.g., Herman and Strohmer [24], Herman and Needell [25],

and Chi, Pezeshki, Scharf, and Calderbank [26]. Herman et al. analyze the effect of additive perturbations on the Basis Pursuit and CoSaMP algorithms, respectively, whereas Chi et al. analyze the effect, on Basis Pursuit, of a multiplicative basis mismatch matrix that takes the form of the identity matrix plus a perturbation. In [24–26], the authors study the worst-case effects on established algorithms, but stop short of proposing new algorithms.

We are aware of only a few algorithms that explicitly address measurement-matrix uncertainty, all of which consider the additive uncertainty model $\boldsymbol{A} = \widehat{\boldsymbol{A}} + \boldsymbol{E}$, where $\widehat{\boldsymbol{A}}$ is known and $\boldsymbol{E}$ is an unknown perturbation, yielding the observations

$$\boldsymbol{y} = (\widehat{\boldsymbol{A}} + \boldsymbol{E})\boldsymbol{x} + \boldsymbol{w}. \tag{3.1}$$

In [27], Zhu et al. develop the Sparsity-cognizant Total Least Squares (S-TLS) approach, which extends the classical TLS approach (widely applied in the context of $\ell_2$ regularization) to $\ell_1$ regularization, yielding

$$\{\widehat{\boldsymbol{x}}_{\text{S-TLS}}, \widehat{\boldsymbol{E}}_{\text{S-TLS}}\} = \\ \arg\min_{\boldsymbol{x},\boldsymbol{E}} \|(\widehat{\boldsymbol{A}} + \boldsymbol{E})\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda_E \|\boldsymbol{E}\|_F^2 + \lambda\|\boldsymbol{x}\|_1. \tag{3.2}$$

In [28], Rosenbaum and Tsybakov propose the MU-Selector, a modified version of the Dantzig selector [29], which reads

$$\{\widehat{\boldsymbol{x}}_{\text{MU-Selector}}\} = \\ \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \text{ s. t. } \|\widehat{\boldsymbol{A}}^H(\boldsymbol{y} - \widehat{\boldsymbol{A}}\boldsymbol{x})\|_\infty \leq \lambda\|\boldsymbol{x}\|_1 + \epsilon. \tag{3.3}$$

The above criteria assume relatively little about the structure of the perturbations $\boldsymbol{w}$ and $\boldsymbol{E}$, and thus obtain algorithms with wide applicability, but—as we shall see—limited performance. In [27], Zhu et al. also proposed a Weighted S-TLS (WS-TLS)

that can exploit structure in the matrix uncertainty $\boldsymbol{E}$ and perform significantly better than S-TLS.

In this chapter, we address sparse-signal recovery under matrix uncertainty in a Bayesian framework with informative priors. In particular, we extend the Approximate Message Passing (AMP) algorithm recently proposed by Donoho, Maleki, and Montanari [3]—and in particular the Generalized AMP (GAMP) proposed by Rangan [6] and reviewed in Chapter 2—to the case of probabilistic uncertainty on the elements of the measurement matrix $\boldsymbol{A}$. Initially, we treat the entries of $\boldsymbol{A}$ as independent random variables that are known only in mean and variance, which can both vary across the entries. The resulting Matrix-Uncertain GAMP (MU-GAMP) provides a computationally efficient way to obtain nearly minimum-mean-squared-error (MMSE) estimates of the unknown signal $\boldsymbol{x}$ in the presence of uncertainties in both the linear matrix transformation $\boldsymbol{A}$ as well as the observations of the transformed outputs $\boldsymbol{Ax}$.

We then turn our attention to parametric matrices of the form $\boldsymbol{A}(\boldsymbol{\theta}) = \boldsymbol{A}_0 + \sum_{p=1}^{P} \theta_p \boldsymbol{A}_p$, where $\{\boldsymbol{A}_p\}$ are known and $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_P]^{\mathsf{T}}$ unknown. We then propose a scheme that alternates between the estimation of $\boldsymbol{\theta}$ and the estimation of $\boldsymbol{x}$. Conveniently, both estimation steps can be performed using the already developed MU-GAMP framework. A salient feature of this approach is that we alternate soft estimates as opposed to point estimates.

Throughout the chapter, we use boldface capital letters to denote matrices and boldface small letters to denote vectors, $\boldsymbol{I}$ and $\boldsymbol{0}$ to denote the identity matrix and zero matrices, $(\cdot)^{\mathsf{T}}$ transpose, and $(\cdot)^*$ conjugate. For $x_j$ a realization of random variable $X_j$, we use $\mathrm{E}_{X_j}\{x_j\}$ to denote mean, $\mathrm{var}_{X_j}\{x_j\}$ variance, $p_{X_j}(x_j)$ the pdf, and

$p_{X_j|D_j}(x_j \,|\, d_j)$ the pdf conditioned on $D_j = d_j$, and sometimes we omit the subscript when there is no danger of confusion. To denote the Gaussian pdf with mean $\widehat{x}$ and variance $\nu^x$, we use $\mathcal{N}(x; \widehat{x}, \nu^x)$.

## 3.2 A Large-System Blessing?

Before getting into the details of MU-GAMP, we make a curious observation: As the problem dimensions grow large, the effect of *uniform* matrix uncertainty is identical to additive white Gaussian noise (AWGN) on the observations. The following proposition makes our claim precise.

**Proposition 3.2.1** *Consider an $M$-dimensional observation of the form in (3.1), written equivalently as*

$$\boldsymbol{y} = \widehat{\boldsymbol{A}}\boldsymbol{x} + \boldsymbol{e} + \boldsymbol{w} \quad \textit{for} \quad \boldsymbol{e} \triangleq \boldsymbol{E}\boldsymbol{x}. \tag{3.4}$$

*Suppose that $N$-dimensional $\boldsymbol{x}$ is $K$-sparse, and that the matrix uncertainty $\boldsymbol{E}$ is* uniform*, i.e., $\{E_{mn}\}$ are i.i.d zero-mean random variables with variance $\nu^E = c^E/M$ for bounded positive $c^E$ (but otherwise arbitrary distribution). In the large-system limit (i.e., $M, N, K \to \infty$ with fixed $\delta \triangleq M/N$ and $\rho \triangleq K/M$), the additive "interference" $\boldsymbol{e}$ becomes i.i.d zero-mean Gaussian with variance $\nu^e = c^E \delta^{-1} \|\boldsymbol{x}\|_2^2 / N$.*

*Proof.* Since the rows of $\boldsymbol{E}$ are statistically independent, the elements $\{e_m\}$ of $\boldsymbol{e}$ are independent as well. Moreover, $e_m = \sum_{k=1}^{K} E_{m,n(k)} x_{n(k)}$, where $n(k)$ indexes the $k^{th}$ non-zero element of $\boldsymbol{x}$. Thus, in the large-system limit (i.e., $K \to \infty$), the central limit theorem implies that $e_m$ is zero-mean Gaussian with variance $\nu^e \triangleq \nu^E \|\boldsymbol{x}\|_2^2 = c^E \delta^{-1} \|\boldsymbol{x}\|_2^2 / N$. □

The implication of Proposition 3.2.1 is that, for problems of *uniform* matrix uncertainty and *suitably large* dimension, there is no need to design new algorithms that handle matrix uncertainty; those designed to handle AWGN (e.g., LASSO [30], GAMP, etc.) suffice, so long as they are properly tuned to handle the additional AWGN power $\nu^e$.

Now, whether or not the large-system behavior predicted by Proposition 3.2.1 manifests at a given *finite* $(M, N, K)$ depends on the distribution of i.i.d $\{E_{mn}\}$ and the sparsity $K$. If $\{E_{mn}\}$ are far from Gaussian (e.g., sparse) and $K$ is relatively small, the distribution of $\{e_m\}$ can be far from Gaussian. On the other hand, if $\{E_{mn}\}$ is Gaussian, then $e_m$ will also be Gaussian, for any $K$.

Although, to our knowledge, Proposition 3.2.1 is novel[8], the empirical results in previous works support its claim; see, e.g., the negligible difference between optimally tuned versions of S-TLS and LASSO under i.i.d Gaussian $\boldsymbol{E}$ in [27, Fig. 3]. In Section 3.3.3, we will provide further empirical support.

## 3.3 Matrix-Uncertain GAMP

### 3.3.1 Background on GAMP

In the Bayesian approach to compressed sensing, it is typically presumed that the signal $\boldsymbol{x}$ is drawn from a known separable pdf $p(\boldsymbol{x}) = \prod_n p_X(x_n)$, where $p_X(.)$ promotes sparsity or compressibility. Similarly, the noise $\boldsymbol{w}$ is drawn from a known separable pdf $p(\boldsymbol{w}) = \prod_m p_W(w_m)$. Given the observations $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{w}$, one would

[8]For a recent study of this problem, which appeared after our work on MU-GAMP was published in [7], see [31].

ideally like to compute the full joint posterior $p(\boldsymbol{x} \,|\, \boldsymbol{y})$. This is, however, not tractable for the pdfs and problem dimensions typical in compressed sensing. Thus, one often settles for approximate MAP or MMSE estimates.

The original AMP algorithm [3] assumes Laplacian $p_X(.)$ and Gaussian $p_W(.)$, and seeks the MAP solution using an approximation of loopy belief propagation. The approximation, which becomes tight in the large-system limit, is based on the CLT and Taylor-series expansions, and relies on the elements of $\boldsymbol{A}$ to be known realizations of an independent zero-mean $1/M$-variance random variable.

Rangan proposed a Generalized AMP (GAMP) [6] that 1) handles either MAP or MMSE, 2) allows arbitrary $A_{mn}$, 3) allows an arbitrary signal distribution $p_X(.)$, and 4) allows an arbitrary separable pdf $p(\boldsymbol{y} \,|\, \boldsymbol{z}) = \prod_m p_{Y|Z}(y_m \,|\, z_m)$ relating the observations $\boldsymbol{y}$ to the linearly transformed outputs $\boldsymbol{z} \triangleq \boldsymbol{A}\boldsymbol{x}$. This observation-uncertainty model subsumes the case of additive noise $\boldsymbol{w}$ with arbitrary distribution $p_W(.)$ via $p_{Y|Z}(y_m \,|\, z_m) = p_W(y_m - z_m)$, but also handles nonlinear output transformations like that used in logistic regression.

### 3.3.2 Matrix-Uncertain GAMP

We now propose a Matrix-Uncertain GAMP (MU-GAMP) that extends GAMP [6] to the case of uncertainty in the measurement matrix $\boldsymbol{A}$. Unlike GAMP, which treats $\{A_{mn}\}$ as fixed and known, MU-GAMP treats $\{A_{mn}\}$ as independent random variables with known mean and variance,

$$\widehat{A}_{mn} = \mathrm{E}\{A_{mn}\} \tag{3.5}$$

$$\nu^A_{mn} = \mathrm{var}\{A_{mn}\}, \tag{3.6}$$

| | | | |
|---|---|---|---|
| definitions: | | | |
| $p_{Z\|Y}(z\|y;\widehat{z},\nu^z)$ | $=$ | $\frac{p_{Y\|Z}(y\|z)\,\mathcal{N}(z;\widehat{z},\nu^z)}{\int_{z'} p_{Y\|Z}(y\|z')\,\mathcal{N}(z';\widehat{z},\nu^z)}$ | (D1) |
| $g_{\mathsf{out}}(y,\widehat{z},\nu^z)$ | $=$ | $\frac{1}{\nu^z}\big(\mathrm{E}_{Z\|Y}\{z\|y;\widehat{z},\nu^z\}-\widehat{z}\big)$ | (D2) |
| $g'_{\mathsf{out}}(y,\widehat{z},\nu^z)$ | $=$ | $\frac{1}{\nu^z}\Big(\frac{\mathrm{var}_{Z\|Y}\{z\|y;\widehat{z},\nu^z\}}{\nu^z}-1\Big)$ | (D3) |
| $p_{X\|\boldsymbol{Y}}(x\|\boldsymbol{y};\widehat{r},\nu^r)$ | $=$ | $\frac{p_X(x)\,\mathcal{N}(x;\widehat{r},\nu^r)}{\int_{x'} p_X(x')\,\mathcal{N}(x';\widehat{r},\nu^r)}$ | (D4) |
| $g_{\mathsf{in}}(\widehat{r},\nu^r)$ | $=$ | $\int_x x\,p_{X\|\boldsymbol{Y}}(x\|\boldsymbol{y};\widehat{r},\nu^r)$ | (D5) |
| $g'_{\mathsf{in}}(\widehat{r},\nu^r)$ | $=$ | $\frac{1}{\nu^r}\int_x \|x-g_{\mathsf{in}}(\widehat{r},\nu^r)\|^2\,p_{X\|\boldsymbol{Y}}(x\|\boldsymbol{y};\widehat{r},\nu^r)$ | (D6) |
| initialize: | | | |
| $\forall n:\widehat{x}_n(1)$ | $=$ | $\int_x x\,p_X(x)$ | (I1) |
| $\forall n:\nu^x_n(1)$ | $=$ | $\int_x \|x-\widehat{x}_n(1)\|^2 p_X(x)$ | (I2) |
| $\forall m:\widehat{u}_m(0)$ | $=$ | $0$ | (I3) |
| for $t=1,2,3,\ldots$ | | | |
| $\forall m:\widehat{z}_m(t)$ | $=$ | $\sum_{n=1}^N \widehat{A}_{mn}\widehat{x}_n(t)$ | (R1) |
| $\forall m:\nu^z_m(t)$ | $=$ | $\sum_{n=1}^N \|\widehat{A}_{mn}\|^2\nu^x_n(t)$ | (R2a) |
| $\forall m:\nu^p_m(t)$ | $=$ | $\nu^z_m(t)+\sum_{n=1}^N \nu^A_{mn}\left(\nu^x_n+\|\widehat{x}_n(t)\|^2\right)$ | **(R2b)** |
| $\forall m:\widehat{p}_m(t)$ | $=$ | $\widehat{z}_m(t)-\nu^z_m(t)\,\widehat{u}_m(t-1)$ | (R3) |
| $\forall m:\widehat{u}_m(t)$ | $=$ | $g_{\mathsf{out}}(y_m,\widehat{p}_m(t),\nu^p_m(t))$ | (R4) |
| $\forall m:\nu^u_m(t)$ | $=$ | $-g'_{\mathsf{out}}(y_m,\widehat{p}_m(t),\nu^p_m(t))$ | (R5) |
| $\forall n:\nu^r_n(t)$ | $=$ | $\big(\sum_{n=1}^N \|\widehat{A}_{mn}\|^2\nu^u_m(t)\big)^{-1}$ | (R6) |
| $\forall n:\widehat{r}_n(t)$ | $=$ | $\widehat{x}_n(t)+\nu^r_n(t)\sum_{m=1}^M \widehat{A}^*_{mn}\widehat{u}_m(t)$ | (R7) |
| $\forall n:\nu^x_n(t+1)$ | $=$ | $\nu^r_n(t)g'_{\mathsf{in}}(\widehat{r}_n(t),\nu^r_j(t))$ | (R8) |
| $\forall n:\widehat{x}_n(t+1)$ | $=$ | $g_{\mathsf{in}}(\widehat{r}_n(t),\nu^r_n(t))$ | (R9) |
| end | | | |

Table 3.1: The MU-GAMP Algorithm

reducing to GAMP in the case that $\nu^A_{mn} = 0$. Note that, with $\boldsymbol{E} \triangleq \boldsymbol{A} - \widehat{\boldsymbol{A}}$, we recover exactly the perturbation model $\boldsymbol{A} = \widehat{\boldsymbol{A}} + \boldsymbol{E}$ used in (3.1), but now with the implicit assumption that $E_{mn}$ has zero mean and variance $\nu^A_{mn}$.

The derivation of MU-GAMP is very similar to the GAMP derivation provided in Section 2.4. For the sake of brevity, we avoid repeating the modified derivation here. The resulting algorithm is given in Table 3.1,[9] where the only difference from the original GAMP is the additional step (R2b). With this step, MU-GAMP requires an additional matrix multiply, although the cost of this multiplication may be reduced when $\nu^A_{mn}$ is structured. For example, when $\nu^A_{mn} = \nu^A_m \; \forall n$, the matrix multiplication in (R2b) reduces to a sum.

### 3.3.3 Empirical Study

We now study empirical performance under *uniform* and *non-uniform* matrix uncertainty. In both cases, we plot Normalized Mean Squared Error (NMSE) versus $M/N$ at $N = 256$ and $K/M = 0.2$, where the relatively small problem size was used due to the implementation complexity of the MU-Selector. The $K$ non-zero entries of the signal $\boldsymbol{x}$ were drawn $\pm 1$ with equal probability, the (known) matrix means $\{\widehat{A}_{mn}\}$ were i.i.d $\mathcal{N}(0, 1/M)$, and the noise $\boldsymbol{w}$ was i.i.d $\mathcal{N}(0, \nu^w)$.

To illustrate the effect of *uniform* matrix uncertainty, we drew the matrix errors $\{E_{mn}\}$ i.i.d $\mathcal{N}(0, \nu^E)$, noting that in this case $\boldsymbol{e} = \boldsymbol{E}\boldsymbol{x}$ is truly i.i.d Gaussian (for any given $\boldsymbol{x}$). Moreover, we set $\nu^E = \nu^w$ such that the signal to interference-plus-noise ratio (SINR) $\mathrm{E}\{\|\widehat{\boldsymbol{A}}\boldsymbol{x}\|_2^2\} / \mathrm{E}\{\|\boldsymbol{e}+\boldsymbol{w}\|_2^2\} = 20$ dB. Under this setup, we ran MU-GAMP under the true (uniform) matrix error variance $\nu^A_{mn} = \nu^E$, the true noise statistics, the true signal variance and sparsity rate, but a (mismatched) Bernoulli-Gaussian

[9] GAMPmatlab [19] includes the MU extension.

signal pdf. We also ran the original GAMP under the same signal prior and the compensated AWGN variance $\nu^e + \nu^w$, for $\nu^e \triangleq \text{var}\{e_m\} = K\nu^E$. We then ran S-TLS, the MU-Selector, and LASSO (via SpaRSA [32]), each debiased and with "genie-aided" tuning: for each realization, each algorithm was run under several values of its tuning parameter, and the tuning yielding minimal NMSE was selected.

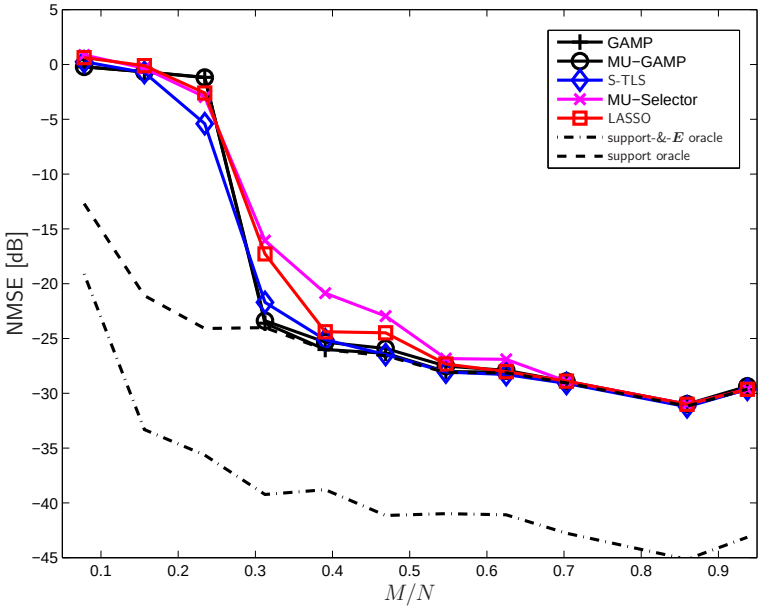


Figure 3.1: 10-trial median NMSE under uniform matrix error variance $\nu^E$.

Figure 3.1 shows the resulting NMSE performance of each algorithm, as well as that of two oracle estimators: support-aware LMMSE, and support-and-$\boldsymbol{E}$-aware LMMSE. We note that, under a Bernoulli-Gaussian signal pdf, the NMSEs of GAMP and MU-GAMP are lower bounded by these respective oracles. The figure shows that GAMP and MU-GAMP yield essentially identical NMSE, and that for $M/N > 0.3$, this NMSE essentially coincides with the support-oracle bound. Meanwhile, the

debiased and genie-tuned incarnations of S-TLS, the MU-Selector, and LASSO show performance that is only slightly worse than GAMP and MU-GAMP for $M/N > 0.3$. The fact that the matrix-uncertain algorithms (i.e., MU-GAMP, S-TLS, MU-Selector) and the standard algorithms (i.e., GAMP, LASSO) perform near-identically under *uniform* matrix uncertainty supports the claim of Proposition 3.2.1.

Next, we examine the effect of *non-uniform* matrix uncertainty. For this, we used the same setup as in the previous experiment, except that we used *non-uniform* variances $\{\nu^E_{mn}\}$ such that $\nu^E_{mn} = 0$ for 99% of the entries, while $\nu^E_{mn} = C^E$ for the remaining 1% of the entries, where $C^E$ was chosen to make the cumulative error $\nu^e$ identical to the previous experiment. MU-GAMP was then run under the true (now non-uniform) $\nu^A_{mn} = \nu^E_{mn}$, while GAMP was run under the compensated AWGN variance $\nu^e + \nu^w$, as before. We also implemented the Weighted S-TLS (WS-TLS) from [27], which was given knowledge of the non-uniform $\{\nu^E_{mn}\}$.

Figure 3.2 shows the resulting NMSE. In the figure, we see that the algorithms assuming uniform matrix uncertainty $\nu^E$ (i.e., S-TLS and the MU-Selector) perform essentially the same in this experiment as they did in the previous experiment, which is due to the fact that $\nu^e$ was calibrated across experiments. Furthermore, these algorithms do essentially no better than those designed for AWGN (i.e., LASSO and GAMP), which makes sense in light of Proposition 3.2.1. However, the algorithms exploiting non-uniform uncertainty $\{\nu^E_{mn}\}$ (i.e., WS-TLS and MU-GAMP) do significantly better. In fact, MU-GAMP performs quite close to the support-and-$\boldsymbol{E}$-aware oracle bound for $M/N > 0.3$.

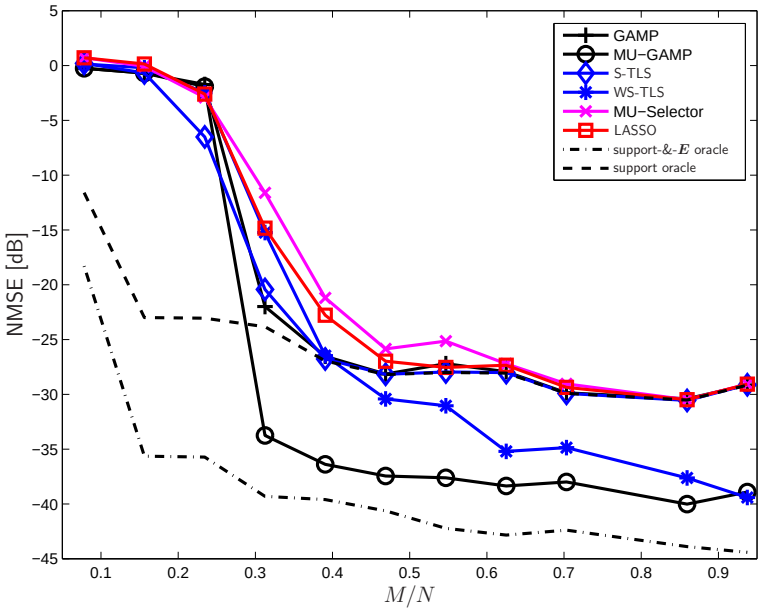


Figure 3.2: 10-trial median NMSE under non-uniform error variance $\{\nu^E_{mn}\}$.

## 3.4 Alternating MU-GAMP

The performance of any reasonable compressive-sensing algorithm will improve as matrix uncertainty diminishes, and one way to reduce uncertainty is to explicitly estimate the unknown matrix $\boldsymbol{A}$. In fact, this is the goal of Dictionary Learning [33], where a large number of measurement vectors $\{\boldsymbol{y}_t\}_{t=1}^T$ are assumed to be available. Since we are interested in estimating $\boldsymbol{A}$ from one (or very few) measurement vectors, we consider structured forms of $\boldsymbol{A}$ that depend on only a few parameters $\boldsymbol{\theta} \in \mathbb{C}^P$. In particular, we consider affine linear[10] models of the form (noting similarities to [27])

$$\boldsymbol{A}(\boldsymbol{\theta}) = \boldsymbol{A}_0 + \textstyle\sum_{p=1}^P \theta_p \boldsymbol{A}_p \tag{3.7}$$

[10]The affine linear model (3.7) may arise from a first-order Taylor series approximation of a non-linear model $\boldsymbol{A}(\boldsymbol{\theta})$ around the point $\widehat{\boldsymbol{\theta}}$, in which case $\boldsymbol{A}_0 = \boldsymbol{A}(\widehat{\boldsymbol{\theta}})$ and $\boldsymbol{A}_p = \partial \boldsymbol{A}(\boldsymbol{\theta})/\partial\theta_p|_{\boldsymbol{\theta}=\widehat{\boldsymbol{\theta}}}$.

with known $\{\boldsymbol{A}_p\}_{p=0}^P$ and unknown $\boldsymbol{\theta}$. Several examples of this structure are discussed in the sequel. Moreover, (3.7) handles the case of *un*structured $\boldsymbol{A}$ via $P = MN$, $\boldsymbol{A}_0 = \boldsymbol{0}$, and $\{\boldsymbol{A}_p\}_{p=1}^P$ each containing a single distinct non-zero entry.

### 3.4.1 Alternating MU-GAMP

We now propose a scheme to jointly estimate $\{\boldsymbol{x}, \boldsymbol{\theta}\}$ based on the previously developed MU-GAMP. The proposed scheme is an iterative one that alternates between the estimation of $\boldsymbol{x}$ and $\boldsymbol{\theta}$. Say the mean and variance of $\theta_p$ are given by $\widehat{\theta}_p$ and $\nu_p^\theta$, respectively. Then it holds that

$$\widehat{A}_{mn} \triangleq \mathrm{E}\{A_{mn}(\boldsymbol{\theta})\} = A_{0,mn} + \textstyle\sum_{p=1}^P \widehat{\theta}_p A_{p,mn} \tag{3.8}$$

$$\nu_{mn}^A \triangleq \mathrm{var}\{A_{mn}(\boldsymbol{\theta})\} = \textstyle\sum_{p=1}^P \nu_p^\theta |A_{p,mn}|^2, \tag{3.9}$$

where $A_{p,mn}$ denotes the $m^{th}$ row and $n^{th}$ column of $\boldsymbol{A}_p$. Thus, given the soft parameter estimates $(\widehat{\boldsymbol{\theta}}, \boldsymbol{\nu}^\theta)$, one can directly compute the matrix uncertainty statistics $\{\widehat{A}_{mn}\}$ and $\{\nu_{mn}^A\}$, and—with them—run MU-GAMP to estimate the signal vector $\boldsymbol{x}$, which will produces the marginal posterior mean and variance vectors $(\widehat{\boldsymbol{x}}, \boldsymbol{\nu}^x)$.

Then, given the soft signal estimates $(\widehat{\boldsymbol{x}}, \boldsymbol{\nu}^x)$, we can update the parameter means and variances $(\widehat{\boldsymbol{\theta}}, \boldsymbol{\nu}^\theta)$, also using MU-GAMP. To see how, we first notice that the linear outputs $\boldsymbol{z}$ in the GAMP observation model $p(\boldsymbol{y} \,|\, \boldsymbol{z})$ take the form

$$\boldsymbol{z} = \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{x} = \boldsymbol{A}_0\boldsymbol{x} + \textstyle\sum_{p=1}^P \boldsymbol{A}_p\boldsymbol{x}\,\theta_p = \boldsymbol{B}(\boldsymbol{x})\boldsymbol{\theta} \tag{3.10}$$

for $\boldsymbol{\theta} \triangleq [\theta_0, \theta_1, \ldots, \theta_P]^\mathsf{T}$, $\theta_0 \triangleq 1$, and the (uncertain) matrix

$$\boldsymbol{B}(\boldsymbol{x}) \triangleq \left[\; \boldsymbol{A}_0\boldsymbol{x} \,\middle|\, \boldsymbol{A}_1\boldsymbol{x} \,\middle|\, \cdots \,\middle|\, \boldsymbol{A}_P\boldsymbol{x} \;\right]. \tag{3.11}$$

Given $(\widehat{\boldsymbol{x}}, \boldsymbol{\nu}^x)$, the mean and variance of $B_{mp}$ are simply

$$\widehat{B}_{mp} \triangleq \mathrm{E}\{B_{mp}(\boldsymbol{x})\} = \textstyle\sum_{n=1}^{N} A_{p,mn}\widehat{x}_n \tag{3.12}$$

$$\nu^B_{mp} \triangleq \mathrm{var}\{B_{mp}(\boldsymbol{x})\} = \textstyle\sum_{n=1}^{N} |A_{p,mn}|^2\nu^x_n, \tag{3.13}$$

which, together with an appropriate prior pdf on $\{\theta_p\}$, are the ingredients needed to estimate $\boldsymbol{\theta}$ with MU-GAMP, yielding updated soft outputs $(\widehat{\boldsymbol{\theta}}, \boldsymbol{\nu}^\theta)$. For example, if $\{\theta_p\}_{p=1}^P$ were known to be sparse, then a sparsifying prior would be appropriate. For $\theta_0$, a prior with all mass at 1 would suffice to handle the constraint $\theta_0 = 1$.

Alternating between these two MU-GAMP steps, we can obtain successively refined estimates of $(\widehat{\boldsymbol{x}}, \boldsymbol{\nu}^x)$ and $(\widehat{\boldsymbol{\theta}}, \boldsymbol{\nu}^\theta)$. Each MU-GAMP step itself involves several iterations, but relatively few would be needed if they were "warm started" at the values of the previous estimates. Note that, unlike typical iterative schemes for dictionary learning [33], which alternate between point estimates, ours alternate between *soft* estimates, i.e., mean/variance pairs.

### 3.4.2 Empirical Study

We now present three empirical experiments that investigate MU-GAMP and alternating MU-GAMP (A-MU-GAMP) under parametric matrix uncertainty. In all cases, we used $M = 103$, $N = 256$, i.i.d Gaussian $\boldsymbol{A}_0 \in \mathbb{C}^{M\times N}$ and $\boldsymbol{\theta} \in \mathbb{C}^P$, i.i.d Bernoulli-Gaussian $\boldsymbol{x} \in \mathbb{C}^N$ with $K = 20$, and complex AWGN. MU-GAMP used the apriori matrix statistics $\{\widehat{A}_{mn}, \nu^A_{mn}\}$ from (3.8)–(3.9). A-MU-GAMP was initialized with the same statistics, but was able to reduce the variances $\{\nu^A_{mn}\}$ through several iterations.

First, we study the role of matrix-uncertainty dimension $P$ on the NMSE performance of MU-GAMP and A-MU-GAMP. For this example, we used i.i.d Gaussian

$\{\boldsymbol{A}_p\}_{p=1}^P$. As $P$ was varied, $\{\nu_p^\theta\}$ was normalized to fix the energy of the uncertainty term $\boldsymbol{E} = \sum_{p=1}^P \theta_p \boldsymbol{A}_p$ such that the overall SINR = 20 dB (as in Figs. 3.1–3.2). Fig. 3.3 shows the resulting NMSE-versus-$P$, where—as expected—MU-GAMP maintains a constant performance versus $P$, whereas A-MU-GAMP benefits when $P$ is small (and thus $\boldsymbol{\theta}$ can be learned).

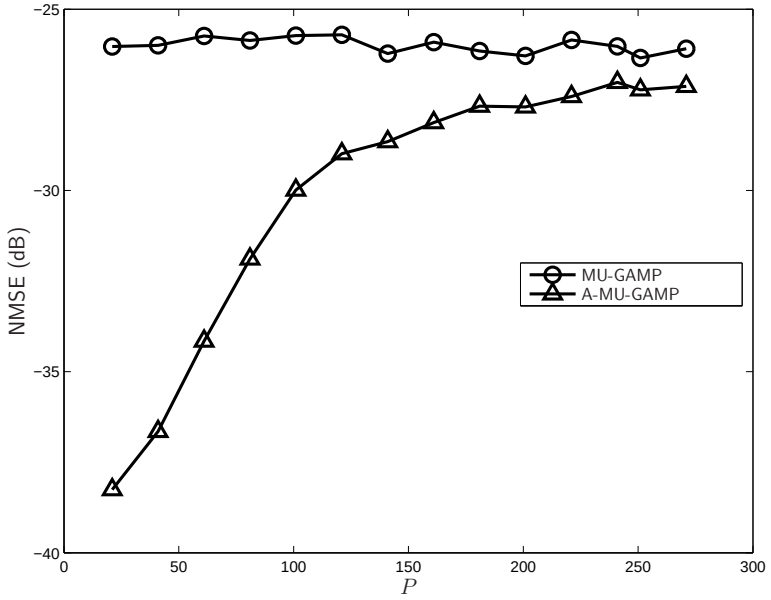


Figure 3.3: 10-trial median NMSE for estimation of $\boldsymbol{x}$ versus the parametric matrix-uncertainty dimension $P$.

Next, we consider a *channel-calibration* example involving $P = 10$ parallel linear measurement "channels", each with an unknown offset. For this, we constructed each matrix $\{\boldsymbol{A}_p\}_{p=1}^P$ to have ones in $1/P$ of its rows and zeros elsewhere, so that i.i.d Gaussian $\theta_p$ modeled the additive error in the $p^{th}$ channel. Here, $\nu^w$ and $\nu^\theta$

were set so that $\mathrm{E}\{\|\boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{x}\|_2^2\}/\,\mathrm{E}\{\|\boldsymbol{w}\|_2^2\} = 20$ dB. Figure 3.4 shows that A-MU-GAMP approaches the performance of $\boldsymbol{\theta}$-aware GAMP when estimating $\boldsymbol{x}$, which comes within 2 dB of the support-and-$\boldsymbol{\theta}$-aware oracle MMSE. The star shows the NMSE of MU-GAMP, which is about 20 dB worse. Meanwhile, when estimating $\boldsymbol{\theta}$, A-MU-GAMP approaches the performance of $\boldsymbol{x}$-aware GAMP.

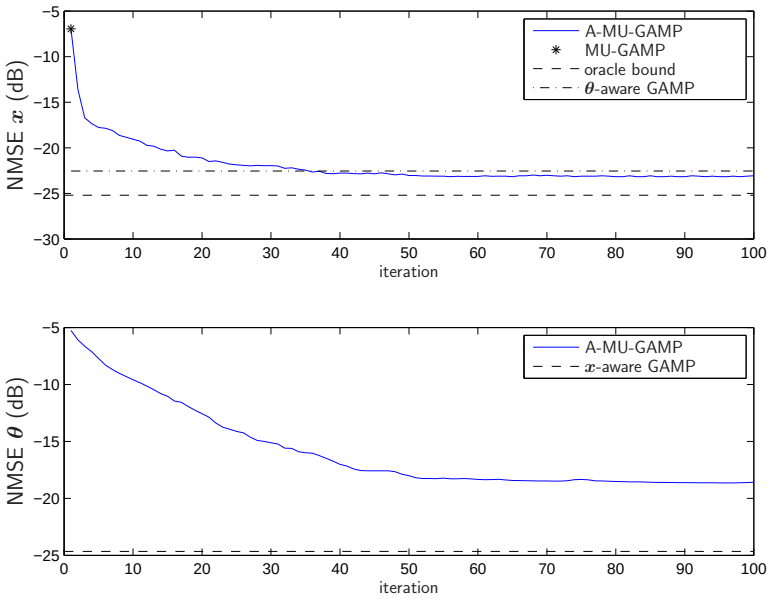


Figure 3.4: 100-trial median NMSE of A-MU-GAMP when iteratively estimating $\boldsymbol{x}$ and $\boldsymbol{\theta}$ in the channel calibration example.

Finally, we consider a *compressive blind-deconvolution* example. Here, $\boldsymbol{A}(\boldsymbol{\theta}) = \boldsymbol{\Phi}\,\mathcal{C}(\boldsymbol{\theta})$ where $\mathcal{C}(\boldsymbol{\theta})$ is circulant with first column $\boldsymbol{\theta} \in \mathbb{C}^N$ and $\boldsymbol{\Phi} = \begin{bmatrix}\boldsymbol{I}_M & \boldsymbol{0}\end{bmatrix}$. As before, $\nu^w$ ensured $\mathrm{E}\{\|\boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{x}\|_2^2\}/\,\mathrm{E}\{\|\boldsymbol{w}\|_2^2\} = 20$ dB. Due to the size of the uncertainty dimension, $P = N$, we used $T = 8$ measurement vectors $\{\boldsymbol{y}_t\}_{t=1}^T$, which is still much fewer than typical in dictionary learning. Figure 3.5 demonstrates that, once again,

A-MU-GAMP is able to effectively learn both $\boldsymbol{x}$ and $\boldsymbol{\theta}$ with near-oracle MMSE, doing $\approx 20$ dB better than MU-GAMP.

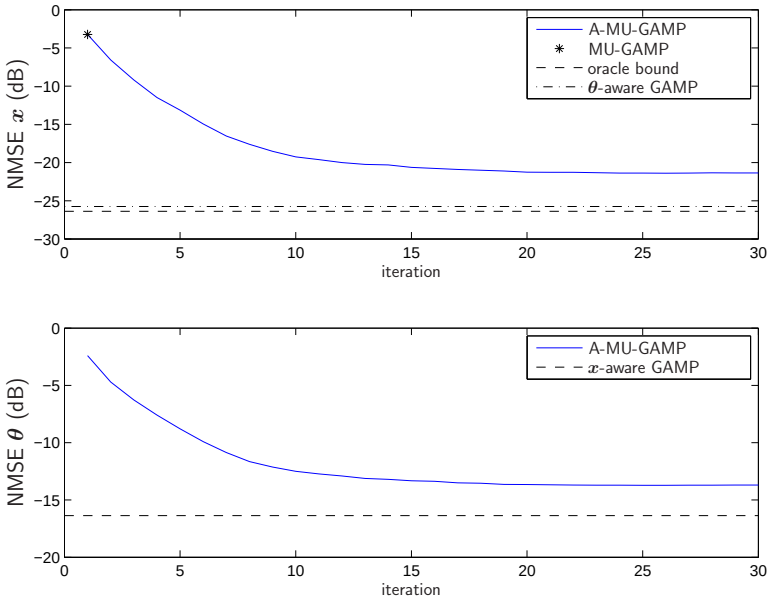


Figure 3.5: 100-trial median NMSE of A-MU-GAMP when iteratively estimating $\boldsymbol{x}$ and $\boldsymbol{\theta}$ in the compressive blind deconvolution example.

## 3.5 Conclusion

In this chapter, we proposed a matrix-uncertainty (MU) extension of the GAMP algorithm, as well as an alternating A-MU-GAMP that aims to recover both the signal and the unknown (possibly parametric) measurement matrix. We also provided theoretical and empirical evidence of the following possibly unexpected fact: as the dimensions grow large, the effect of *uniform* matrix uncertainty reduces to AWGN, and can thus be handled by matrix-certain algorithms. Our MU-GAMP approach

can, however, exploit knowledge of *non*-uniform matrix uncertainty to do significantly better. Moreover, our A-MU-GAMP approach, which exploits soft information (as opposed to point estimates), achieves near-oracle performance.

MU-GAMP does not attempt to estimate the perturbed matrix $\boldsymbol{A}$ itself. While A-MU-GAMP offers a method for estimating this operator for a particular class of parametric models, we pursue a more flexible approach in subsequent chapters. In particular, Chapter 4 develops a *bilinear* version of GAMP that jointly estimates $\boldsymbol{A}$ and the possibly matrix-valued signal $\boldsymbol{X}$. This approach is applicable to dictionary learning, matrix completion, robust principle components analysis (PCA), and other related problems. Furthermore, a version of this algorithm which handles parametric operators similar to those addressed by A-MU-GAMP will be developed in Chapter 5.

# Chapter 4: Bilinear Generalized Approximate Message Passing

## 4.1 Introduction

In this chapter, we present a new algorithmic framework for the following *generalized bilinear* inference problem: estimate the matrices $\boldsymbol{A} = [a_{mn}] \in \mathbb{R}^{M \times N}$ and $\boldsymbol{X} = [x_{nl}] \in \mathbb{R}^{N \times L}$ from a matrix observation $\boldsymbol{Y} \in \mathbb{R}^{M \times L}$ that is statistically coupled to their product, $\boldsymbol{Z} \triangleq \boldsymbol{A}\boldsymbol{X}$. In doing so, we treat $\boldsymbol{A}$ and $\boldsymbol{X}$ as realizations of independent random matrices $\mathsf{A}$ and $\mathsf{X}$ with known separable pdfs (or pmfs in the case of discrete models), i.e.,

$$p_{\mathsf{A}}(\boldsymbol{A}) = \prod_m \prod_n p_{\mathsf{a}_{mn}}(a_{mn}) \tag{4.1}$$

$$p_{\mathsf{X}}(\boldsymbol{X}) = \prod_n \prod_l p_{\mathsf{x}_{nl}}(x_{nl}), \tag{4.2}$$

and we likewise assume that the likelihood function of $\boldsymbol{Z}$ is known and separable, i.e.,

$$p_{\mathsf{Y}|\mathsf{Z}}(\boldsymbol{Y} \mid \boldsymbol{Z}) = \prod_m \prod_l p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml} \mid z_{ml}). \tag{4.3}$$

Recently, various special cases of this problem have gained the intense interest of the research community, e.g.,

1. *Matrix Completion*: In this problem, one observes a few (possibly noise-corrupted) entries of a low-rank matrix and the goal is to infer the missing

entries. In our framework, $\boldsymbol{Z}=\boldsymbol{A}\boldsymbol{X}$ would represent the complete low-rank matrix (with tall $\boldsymbol{A}$ and wide $\boldsymbol{X}$) and $p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}$ the observation mechanism, which would be (partially) informative about $\mathsf{z}_{ml}$ at the observed entries $(m,l)\in\Omega$ and non-informative at the missing entries $(m,l)\notin\Omega$.

2. *Robust PCA*: Here, the objective is to recover a low-rank matrix (or its principal components) observed in the presence of noise and sparse outliers. In our framework, $\boldsymbol{Z}=\boldsymbol{A}\boldsymbol{X}$ could again represent the low-rank matrix, and $p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}$ the noise-and-outlier-corrupted observation mechanism. Alternatively, $\boldsymbol{X}$ could also capture the outliers, as described in the sequel.

3. *Dictionary Learning*: Here, the objective is to learn a dictionary $\boldsymbol{A}$ for which there exists a sparse data representation $\boldsymbol{X}$ such that $\boldsymbol{A}\boldsymbol{X}$ closely matches the observed data $\boldsymbol{Y}$. In our framework, $\{p_{\mathsf{x}_{nl}}\}$ would be chosen to induce sparsity, $\boldsymbol{Z}=\boldsymbol{A}\boldsymbol{X}$ would represent the noiseless observations, and $\{p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}\}$ would model the (possibly noisy) observation mechanism.

While a plethora of approaches to these problems have been proposed based on optimization techniques (e.g., [34–44]), greedy methods (e.g., [45–49]), Bayesian sampling methods (e.g., [50,51]), variational methods (e.g., [52–56]), and discrete message passing (e.g., [57]), ours is based on the *Approximate Message Passing* (AMP) framework, an instance of loopy belief propagation (LBP) [58] that was recently developed to tackle *linear* [3, 4, 16] and *generalized linear* [6] inference problems encountered in the context of compressive sensing (CS). In the generalized-linear CS problem, one estimates $\boldsymbol{x}\in\mathbb{R}^N$ from observations $\boldsymbol{y}\in\mathbb{R}^M$ that are statistically coupled to the

transform outputs $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{x}$ through a separable likelihood function $p_{\mathsf{y}|\mathsf{z}}(\boldsymbol{y}|\boldsymbol{z})$, where in this case the transform $\boldsymbol{A}$ is *fixed and known*.

In the context of CS, the AMP framework yields algorithms with remarkable properties: i) solution trajectories that, in the large-system limit (i.e., as $M, N \to \infty$ with $M/N$ fixed, under iid sub-Gaussian $\boldsymbol{A}$) are governed by a state-evolution whose fixed points—when unique—yield the true posterior means [59, 60] and ii) a low implementation complexity (i.e., dominated by one multiplication with $\boldsymbol{A}$ and $\boldsymbol{A}^{\mathsf{T}}$ per iteration, and relatively few iterations) [16]. Thus, a natural question is whether the AMP framework can be successfully applied to the generalized *bilinear* problem described earlier.

In this chapter, we propose an AMP-based approach to generalized bilinear inference that we henceforth refer to as *Bilinear Generalized AMP* (BiG-AMP), and we uncover special cases under which the general approach can be simplified. In addition, we propose an adaptive damping [18] mechanism, an expectation-maximization (EM)-based [61] method of tuning the parameters of $p_{\mathsf{a}_{mn}}$, $p_{\mathsf{x}_{nl}}$, and $p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}$ (in case they are unknown), and methods to select the rank $N$ (in case it is unknown). In the case that $p_{\mathsf{a}_{mn}}$, $p_{\mathsf{x}_{nl}}$, and/or $p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}$ are completely unknown, they can be modeled as Gaussian-mixtures with mean/variance/weight parameters learned via EM [62]. Finally, we present a detailed numerical investigation of BiG-AMP applied to matrix completion, robust PCA, and dictionary learning. Our empirical results show that BiG-AMP yields an excellent combination of estimation accuracy and runtime when compared to existing state-of-the-art algorithms for each application.

Although the AMP methodology is itself restricted to separable known pdfs (4.1)-(4.3), the results of Part II suggest that this limitation is not an issue for many practical problems of interest. However, in problems where the separability assumption is too constraining, it can be relaxed through the use of hidden (coupling) variables, as originally proposed in the context of "turbo-AMP" [63] and applied to BiG-AMP in [64]. Due to space limitations, however, this approach will not be discussed here. Finally, although we focus on real-valued random variables, all of the methodology described in this work can be easily extended to circularly symmetric complex-valued random variables.

We now discuss related work. One possibility of applying AMP methods to matrix completion was suggested by Montanari in [16, Sec. 9.7.3] but the approach described there differs from BiG-AMP in that it was i) constructed from a factor graph with *vector-valued* variables and ii) restricted to the (incomplete) additive white Gaussian noise (AWGN) observation model. Moreover, no concrete algorithm nor performance evaluation was reported. Since we first reported on BiG-AMP in [65, 66], Rangan and Fletcher [67] proposed an AMP-based approach for the estimation of *rank-one* matrices from AWGN-corrupted observations, and showed that it can be characterized by a state evolution in the large-system limit. More recently, Krzakala, Mézard, and Zdeborová [68] proposed an AMP-based approach to blind calibration and dictionary learning in AWGN that bears similarity to a special case of BiG-AMP, and derived a state-evolution using the cavity method. Their method, however, was not numerically successful in solving dictionary learning problems [68]. The BiG-AMP algorithm that we derive here is a generalization of those in [67, 68] in that it handles *generalized* bilinear observations rather than AWGN-corrupted ones. Moreover,

our work is the first to detail adaptive damping, parameter tuning, and rank-selection mechanisms for AMP based bilinear inference, and it is the first to present an in-depth numerical investigation involving both synthetic and real-world datasets. An application/extension of the BiG-AMP algorithm described here to hyperspectral unmixing (an instance of non-negative matrix factorization) was recently proposed in [64].

The remainder of the chapter is organized as follows. Section 4.2 derives the BiG-AMP algorithm, and Section 4.3 presents several special-case simplifications of BiG-AMP. Section 4.4 describes the adaptive damping mechanism, and Section 4.5 the EM-based tuning of prior parameters and selection of rank $N$. Application-specific issues and numerical results demonstrating the efficacy of our approach for matrix completion, robust PCA, and dictionary learning, are discussed in Sections 4.6–4.8, respectively, and concluding remarks are offered in Section 4.9.

## 4.2 Bilinear Generalized AMP

### 4.2.1 Problem Formulation

For the statistical model (4.1)-(4.3), the posterior distribution is

$$
\begin{aligned}
&p_{\mathbf{X},\mathbf{A}|\mathbf{Y}}(\boldsymbol{X},\boldsymbol{A}\,|\,\boldsymbol{Y}) \\
&= p_{\mathbf{Y}|\mathbf{X},\mathbf{A}}(\boldsymbol{Y}\,|\,\boldsymbol{X},\boldsymbol{A})\,p_{\mathbf{X}}(\boldsymbol{X})\,p_{\mathbf{A}}(\boldsymbol{A})/p_{\mathbf{Y}}(\boldsymbol{Y}) && (4.4)\\
&\propto p_{\mathbf{Y}|\mathbf{Z}}(\boldsymbol{Y}\,|\,\boldsymbol{A}\boldsymbol{X})\,p_{\mathbf{X}}(\boldsymbol{X})\,p_{\mathbf{A}}(\boldsymbol{A}) && (4.5)\\
&= \Big[\prod_m \prod_l p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}\Big(y_{ml}\,\Big|\,\sum_k a_{mk}x_{kl}\Big)\Big] \\
&\quad\times\Big[\prod_n \prod_l p_{\mathsf{x}_{nl}}(x_{nl})\Big]\Big[\prod_m \prod_n p_{\mathsf{a}_{mn}}(a_{mn})\Big], && (4.6)
\end{aligned}
$$

where (4.4) employs Bayes' rule and $\propto$ denotes equality up to a constant scale factor.

The posterior distribution can be represented with a factor graph, as depicted in Fig. 4.1. There, the factors of $p_{\mathbf{X},\mathbf{A}|\mathbf{Y}}$ from (4.6) are represented by "factor nodes" that appear as black boxes, and the random variables are represented by "variable nodes" that appear as white circles. Each variable node is connected to every factor node in which that variable appears. The observed data $\{y_{ml}\}$ are treated as parameters of the $p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}$ factor nodes in the middle of the graph, and not as random variables. The structure of Fig. 4.1 becomes intuitive when recalling that $\mathbf{Z} = \mathbf{AX}$ implies $\mathsf{z}_{ml} = \sum_{n=1}^{N} \mathsf{a}_{mn}\mathsf{x}_{nl}$.

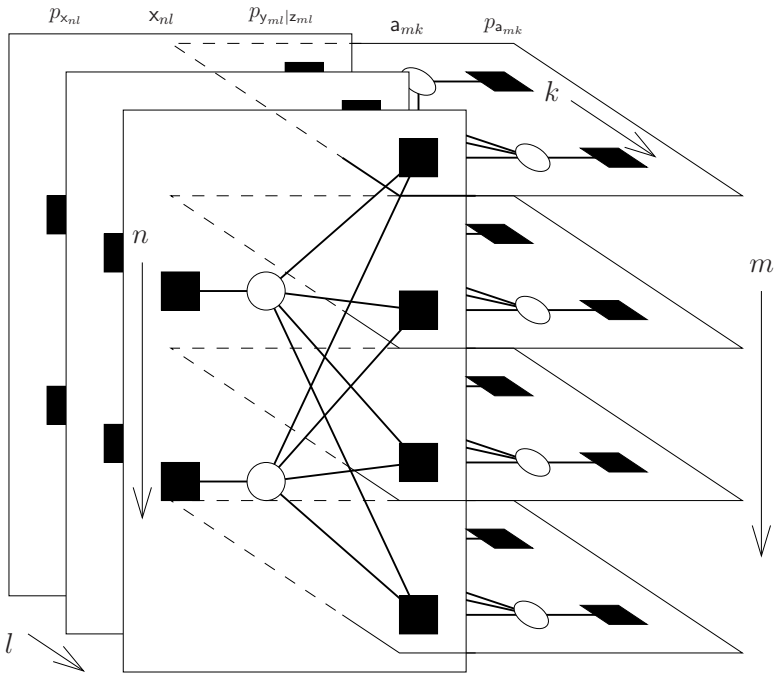


Figure 4.1: The factor graph for generalized bilinear inference for (toy-sized) problem dimensions $M = 4$, $L = 3$, and $N = 2$.

### 4.2.2 Loopy Belief Propagation

In this work, we aim to compute minimum mean-squared error (MMSE) estimates of $\boldsymbol{X}$ and $\boldsymbol{A}$, i.e., the means[11] of the marginal posteriors $p_{\mathsf{x}_{nl}|\mathbf{Y}}(\cdot\,|\,\boldsymbol{Y})$ and $p_{\mathsf{a}_{mn}|\mathbf{Y}}(\cdot\,|\,\boldsymbol{Y})$, for all pairs $(n,l)$ and $(m,n)$. Although exact computation of these quantities is generally prohibitive, they can be efficiently approximated using loopy belief propagation (LBP) [58].

In LBP, beliefs about the random variables (in the form of pdfs or log pdfs) are propagated among the nodes of the factor graph until they converge. The standard way to compute these beliefs, known as the *sum-product algorithm* (SPA) [14, 15], stipulates that the belief emitted by a variable node along a given edge of the graph is computed as the product of the incoming beliefs from all other edges, whereas the belief emitted by a factor node along a given edge is computed as the integral of the product of the factor associated with that node and the incoming beliefs on all other edges. The product of all beliefs impinging on a given variable node yields the posterior pdf for that variable. In cases where the factor graph has no loops, exact marginal posteriors result from two (i.e., forward and backward) passes of the SPA [14,15]. For loopy factor graphs, exact inference is in general NP hard [69] and so LBP does not guarantee correct posteriors. That said, LBP has shown state-of-the-art performance in many applications, such as inference on Markov random fields [70], turbo decoding [71], LDPC decoding [72], multiuser detection [73], and compressive sensing [3, 4, 6, 59, 60].

[11] Another worthwhile objective could be to compute the joint MAP estimate $\arg\max_{\boldsymbol{X},\boldsymbol{A}} p_{\mathbf{X},\mathbf{A}|\mathbf{Y}}(\boldsymbol{X},\boldsymbol{A}\,|\,\boldsymbol{Y})$; we leave this to future work.

In high-dimensional inference problems, exact implementation of the SPA is impractical, motivating approximations of the SPA. A notable example is the *generalized approximate message passing* (GAMP) algorithm, developed in [6] and reviewed in Chapter 2 to solve the generalized CS problem, which exploits the "blessings of dimensionality" that arise when $\boldsymbol{A}$ is a sufficiently large and dense and which was rigorously analyzed in [60]. In the sequel, we derive an algorithm for the generalized bilinear inference BiG-AMP algorithm that employs GAMP-like approximations to the SPA on the factor graph in Fig. 4.1. As we shall see, the approximations are primarily based on central-limit-theorem (CLT) and Taylor-series arguments.

### 4.2.3 Sum-product Algorithm

In our formulation of the SPA, messages take the form of log-pdfs with arbitrary constant offsets. For example, the iteration-$t$ (where $t \in \mathbb{Z}$) message $\Delta^{\mathsf{x}}_{m\to nl}(t,.)$ can be converted to the pdf $\frac{1}{C}\exp(\Delta^{\mathsf{x}}_{m\to nl}(t,.))$, where the choice of scale factor $C = \int_{x_{nl}} \exp(\Delta^{\mathsf{x}}_{m\to nl}(t,x_{nl}))$ ensures that the pdf integrates to one. Four types of message will be used, as specified in Table 4.1. We also find it convenient to express the (iteration-$t$ SPA-approximated) posterior pdfs $p_{\mathsf{x}_{nl}|\mathbf{Y}}(t,.\,|\,\boldsymbol{Y})$ and $p_{\mathsf{a}_{mn}|\mathbf{Y}}(t,.\,|\,\boldsymbol{Y})$ in the log domain as $\Delta^{\mathsf{x}}_{nl}(t,.)$ and $\Delta^{\mathsf{a}}_{mn}(t,.)$, respectively, again with arbitrary constant offsets.

| | |
|---|---|
| $\Delta^{\mathsf{x}}_{m\to nl}(t,.)$ | SPA message from node $p_{\mathsf{y}_{ml}\vert\mathsf{z}_{ml}}$ to node $\mathsf{x}_{nl}$ |
| $\Delta^{\mathsf{x}}_{m\leftarrow nl}(t,.)$ | SPA message from node $\mathsf{x}_{nl}$ to node $p_{\mathsf{y}_{ml}\vert\mathsf{z}_{ml}}$ |
| $\Delta^{\mathsf{a}}_{l\to mn}(t,.)$ | SPA message from node $p_{\mathsf{y}_{ml}\vert\mathsf{z}_{ml}}$ to node $\mathsf{a}_{mn}$ |
| $\Delta^{\mathsf{a}}_{l\leftarrow mn}(t,.)$ | SPA message from node $\mathsf{a}_{mn}$ to node $p_{\mathsf{y}_{ml}\vert\mathsf{z}_{ml}}$ |
| $\Delta^{\mathsf{x}}_{nl}(t,.)$ | SPA-approximated log posterior pdf of $\mathsf{x}_{nl}$ |
| $\Delta^{\mathsf{a}}_{mn}(t,.)$ | SPA-approximated log posterior pdf of $\mathsf{a}_{mn}$ |

Table 4.1: SPA message definitions at iteration $t \in \mathbb{Z}$ for BiG-AMP.

Applying the SPA to the factor graph in Fig. 4.1, we arrive at the following update rules for the four messages in Table 4.1.

$$\begin{aligned}
&\Delta^{\mathsf{x}}_{m\to nl}(t,x_{nl}) \\
&= \log \int_{\{a_{mk}\}_{k=1}^{N},\{x_{rl}\}_{r\neq n}} p_{\mathsf{y}_{ml}\vert\mathsf{z}_{ml}}\Big(y_{ml}\,\Big|\,\sum_{k=1}^{N} a_{mk}x_{kl}\Big) \\
&\quad\times \prod_{r\neq n}\exp\Big(\Delta^{\mathsf{x}}_{m\leftarrow rl}(t,x_{rl})\Big)\prod_{k=1}^{N}\exp\Big(\Delta^{\mathsf{a}}_{l\leftarrow mk}(t,a_{mk})\Big) \\
&\quad+\mathsf{const} \qquad (4.7)
\end{aligned}$$

$$\begin{aligned}
&\Delta^{\mathsf{x}}_{m\leftarrow nl}(t+1,x_{nl}) \\
&= \log p_{\mathsf{x}_{nl}}(x_{nl}) + \sum_{k\neq m}\Delta^{\mathsf{x}}_{k\to nl}(t,x_{nl}) + \mathsf{const} \qquad (4.8)
\end{aligned}$$

$$\begin{aligned}
&\Delta^{\mathsf{a}}_{l\to mn}(t,a_{mn}) \\
&= \log \int_{\{a_{mr}\}_{r\neq n},\{x_{kl}\}_{k=1}^{N}} p_{\mathsf{y}_{ml}\vert\mathsf{z}_{ml}}\Big(y_{ml}\,\Big|\,\sum_{k=1}^{N} a_{mk}x_{kl}\Big) \\
&\quad\times \prod_{k=1}^{N}\exp\Big(\Delta^{\mathsf{x}}_{m\leftarrow kl}(t,x_{kl})\Big)\prod_{r\neq n}\exp\Big(\Delta^{\mathsf{a}}_{l\leftarrow mr}(t,a_{mr})\Big) \\
&\quad+\mathsf{const} \qquad (4.9)
\end{aligned}$$

$$\begin{aligned}
&\Delta^{\mathsf{a}}_{l\leftarrow mn}(t+1,a_{mn}) \\
&= \log p_{\mathsf{a}_{mn}}(a_{mn}) + \sum_{k\neq l}\Delta^{\mathsf{a}}_{k\to mn}(t,a_{mn}) + \mathsf{const}, \qquad (4.10)
\end{aligned}$$

where $\mathsf{const}$ is an arbitrary constant (w.r.t $x_{nl}$ in (4.7) and (4.8), and w.r.t $a_{mn}$ in (4.9) and (4.10)). In the sequel, we denote the mean and variance of the pdf $\frac{1}{C}\exp(\Delta^{\mathsf{x}}_{m\leftarrow nl}(t,.))$ by $\widehat{x}_{m,nl}(t)$ and $\nu^{x}_{m,nl}(t)$, respectively, and we denote the mean and variance of $\frac{1}{C}\exp(\Delta^{\mathsf{a}}_{l\leftarrow mn}(t,.))$ by $\widehat{a}_{l,mn}(t)$ and $\nu^{a}_{l,mn}(t)$. For the log-posteriors, the SPA implies

$$\begin{aligned}&\Delta^{\mathsf{x}}_{nl}(t+1,x_{nl})\\&=\log p_{\mathsf{x}_{nl}}(x_{nl})+\sum_{m=1}^{M}\Delta^{\mathsf{x}}_{m\to nl}(t,x_{nl})+\mathsf{const}\end{aligned}\tag{4.11}$$

$$\begin{aligned}&\Delta^{\mathsf{a}}_{mn}(t+1,a_{mn})\\&=\log p_{\mathsf{a}_{mn}}(a_{mn})+\sum_{l=1}^{L}\Delta^{\mathsf{a}}_{l\to mn}(t,a_{mn})+\mathsf{const},\end{aligned}\tag{4.12}$$

and we denote the mean and variance of $\frac{1}{C}\exp(\Delta^{\mathsf{x}}_{nl}(t,.))$ by $\widehat{x}_{nl}(t)$ and $\nu^{x}_{nl}(t)$, and the mean and variance of $\frac{1}{C}\exp(\Delta^{\mathsf{a}}_{mn}(t,.))$ by $\widehat{a}_{mn}(t)$ and $\nu^{a}_{mn}(t)$.

### 4.2.4 Approximated Factor-to-Variable Messages

We now apply AMP approximations to the SPA updates (4.7)-(4.12). As we shall see, the approximations are based primarily on central-limit-theorem (CLT) and Taylor-series arguments that become exact in the large-system limit, where $M, L, N \to \infty$ with fixed ratios $M/N$ and $L/N$. (Due to the use of finite $M, L, N$ in practice, we still regard them as approximations.) In particular, our derivation will neglect terms that vanish relative to others as $N \to \infty$, which requires that we establish certain scaling conventions. First, we assume w.l.o.g[12] that $\mathrm{E}\{\mathsf{z}^2_{ml}\}$ and $\mathrm{E}\{\mathsf{x}^2_{nl}\}$ scale as $O(1)$, i.e., that the magnitudes of these elements do not change as $N \to \infty$. In this case, assuming that $\mathsf{a}_{mn}$ is zero mean, the relationship $\mathsf{z}_{ml} = \sum_{n=1}^{N} \mathsf{a}_{mn}\mathsf{x}_{nl}$ implies that $\mathrm{E}\{\mathsf{a}^2_{mn}\}$ must scale as $O(1/N)$. These scalings are assumed to hold for

[12]Other scalings on $\mathrm{E}\{\mathsf{z}^2_{ml}\}$, $\mathrm{E}\{\mathsf{x}^2_{nl}\}$, and $\mathrm{E}\{\mathsf{a}^2_{mn}\}$ could be used as long as they are consistent with the relationship $\mathsf{z}_{ml} = \sum_{n=1}^{N} \mathsf{a}_{mn}\mathsf{x}_{nl}$.

| | | | | | |
|---|---|---|---|---|---|
| $\widehat{z}_{ml}(t)$ | $O(1)$ | $\nu^z_{ml}(t)$ | $O(1)$ | $\widehat{x}_{m,nl}(t) - \widehat{x}_{nl}(t)$ | $O(\frac{1}{\sqrt{N}})$ |
| $\widehat{x}_{m,nl}(t)$ | $O(1)$ | $\nu^x_{m,nl}(t)$ | $O(1)$ | $\widehat{x}^2_{m,nl}(t) - \widehat{x}^2_{nl}(t)$ | $O(\frac{1}{\sqrt{N}})$ |
| $\widehat{x}_{nl}(t)$ | $O(1)$ | $\nu^x_{nl}(t)$ | $O(1)$ | $\nu^x_{m,nl}(t) - \nu^x_{nl}(t)$ | $O(\frac{1}{\sqrt{N}})$ |
| $\widehat{a}_{l,mn}(t)$ | $O(\frac{1}{\sqrt{N}})$ | $\nu^a_{l,mn}(t)$ | $O(\frac{1}{N})$ | $\widehat{a}_{l,mn}(t) - \widehat{a}_{mn}(t)$ | $O(\frac{1}{N})$ |
| $\widehat{a}_{mn}(t)$ | $O(\frac{1}{\sqrt{N}})$ | $\nu^a_{mn}(t)$ | $O(\frac{1}{N})$ | $\widehat{a}^2_{l,mn}(t) - \widehat{a}^2_{mn}(t)$ | $O(\frac{1}{N^{3/2}})$ |
| $\widehat{p}_{ml}(t)$ | $O(1)$ | $\nu^p_{ml}(t)$ | $O(1)$ | $\nu^a_{l,mn}(t) - \nu^a_{mn}(t)$ | $O(\frac{1}{N^{3/2}})$ |
| $\widehat{r}_{m,nl}(t)$ | $O(1)$ | $\nu^r_{m,nl}(t)$ | $O(1)$ | $\widehat{r}_{m,nl}(t) - \widehat{r}_{nl}(t)$ | $O(\frac{1}{\sqrt{N}})$ |
| $\widehat{r}_{nl}(t)$ | $O(1)$ | $\nu^r_{nl}(t)$ | $O(1)$ | $\nu^r_{m,nl}(t) - \nu^r_{nl}(t)$ | $O(\frac{1}{N})$ |
| $\widehat{q}_{l,mn}(t)$ | $O(\frac{1}{\sqrt{N}})$ | $\nu^q_{l,mn}(t)$ | $O(\frac{1}{N})$ | $\widehat{q}_{l,mn}(t) - \widehat{q}_{mn}(t)$ | $O(\frac{1}{N})$ |
| $\widehat{q}_{mn}(t)$ | $O(\frac{1}{\sqrt{N}})$ | $\nu^q_{mn}(t)$ | $O(\frac{1}{N})$ | $\nu^q_{l,mn}(t) - \nu^q_{mn}(t)$ | $O(\frac{1}{N^2})$ |
| $\widehat{s}_{ml}(t)$ | $O(1)$ | $\nu^s_{ml}(t)$ | $O(1)$ | | |

Table 4.2: BiG-AMP variable scalings in the large-system limit.

random variables $\mathsf{z}_{ml}$, $\mathsf{a}_{mn}$, and $\mathsf{x}_{ml}$ distributed according to the prior pdfs, according to the pdfs corresponding to the SPA messages (4.7)-(4.10), and according to the pdfs corresponding to the SPA posterior approximations (4.11)-(4.12). These assumptions lead straightforwardly to the scalings of $\widehat{z}_{ml}(t)$, $\nu^z_{ml}(t)$, $\widehat{x}_{m,nl}(t)$, $\nu^x_{m,nl}(t)$, $\widehat{x}_{nl}(t)$, $\nu^x_{nl}(t)$, $\widehat{a}_{l,mn}(t)$, $\nu^a_{l,mn}(t)$, $\widehat{a}_{mn}(t)$, and $\nu^a_{mn}(t)$ specified in Table 4.2. Furthermore, because $\Delta^{\mathsf{x}}_{m\to nl}(t,\cdot)$ and $\Delta^{\mathsf{x}}_{nl}(t,\cdot)$ differ by only one term out of $M$, it is reasonable to assume [6, 16] that the corresponding difference in means $\widehat{x}_{m,nl}(t) - \widehat{x}_{nl}(t)$ and variances $\nu^x_{m,nl}(t) - \nu^x_{nl}(t)$ are both $O(1/\sqrt{N})$, which then implies that $\widehat{x}^2_{m,nl}(t) - \widehat{x}^2_{nl}(t)$ is also $O(1/\sqrt{N})$. Similarly, because $\Delta^{\mathsf{a}}_{l\to mn}(t,\cdot)$ and $\Delta^{\mathsf{a}}_{mn}(t,\cdot)$ differ by only one term out of $N$, where $\widehat{a}_{l,mn}(t)$ and $\widehat{a}_{mn}(t)$ are $O(1/\sqrt{N})$, it is reasonable to assume that $\widehat{a}_{l,mn}(t) - \widehat{a}_{mn}(t)$ is $O(1/N)$ and that both $\nu^a_{l,mn}(t) - \nu^a_{mn}(t)$ and $\widehat{a}^2_{l,mn}(t) - \widehat{a}^2_{mn}(t)$ are $O(1/N^{3/2})$. The remaining entries in Table 4.2 will be explained below.

We start by approximating the message $\Delta^{\mathsf{x}}_{m\to nl}(t,.)$. Expanding (4.7), we find

$$\begin{aligned}
&\Delta^{\mathsf{x}}_{m\to nl}(t,x_{nl}) \\
&= \log \int_{\{a_{mk}\}_{k=1}^{N},\{x_{rl}\}_{r\neq n}} p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}\Big(y_{ml}\,\Big|\, \overbrace{a_{mn}x_{nl}+\sum_{k=1\neq n}^{N} a_{mk}x_{kl}}^{z_{ml}}\Big) \\
&\quad\times \prod_{r\neq n}\exp\Big(\Delta^{\mathsf{x}}_{m\leftarrow rl}(t,x_{rl})\Big)\prod_{k=1}^{N}\exp\Big(\Delta^{\mathsf{a}}_{l\leftarrow mk}(t,a_{mk})\Big) \\
&\quad+\mathsf{const}.
\end{aligned} \tag{4.13}$$

For large $N$, the CLT motivates the treatment of $\mathsf{z}_{ml}$, the random variable associated with the $z_{ml}$ identified in (4.13), conditioned on $\mathsf{x}_{nl}=x_{nl}$, as Gaussian and thus completely characterized by a (conditional) mean and variance. Defining the zero-mean r.v.s $\widetilde{\mathsf{a}}_{l,mn}\triangleq \mathsf{a}_{mn}-\widehat{a}_{l,mn}(t)$ and $\widetilde{\mathsf{x}}_{m,nl}=\mathsf{x}_{nl}-\widehat{x}_{m,nl}(t)$, where $\mathsf{a}_{mn}\sim\frac{1}{C}\exp(\Delta^{\mathsf{a}}_{l\leftarrow mn}(t,\cdot))$ and $\mathsf{x}_{nl}\sim\frac{1}{C}\exp(\Delta^{\mathsf{x}}_{m\leftarrow nl}(t,\cdot))$, we can write

$$\begin{aligned}
\mathsf{z}_{ml} \;=\; &\big(\widehat{a}_{l,mn}(t)+\widetilde{\mathsf{a}}_{l,mn}\big)\mathsf{x}_{nl}+\sum_{k\neq n}\big(\widehat{a}_{l,mk}(t)\,\widehat{x}_{m,kl}(t) \\
&+\widehat{a}_{l,mk}(t)\widetilde{\mathsf{x}}_{m,kl}+\widetilde{\mathsf{a}}_{l,mk}\widehat{x}_{m,kl}(t)+\widetilde{\mathsf{a}}_{l,mk}\widetilde{\mathsf{x}}_{m,kl}\big)
\end{aligned} \tag{4.14}$$

after which it is straightforward to see that

$$\mathrm{E}\{\mathsf{z}_{ml}\,|\,\mathsf{x}_{nl}=x_{nl}\}=\widehat{a}_{l,mn}(t)x_{nl}+\widehat{p}_{n,ml}(t) \tag{4.15}$$

$$\mathrm{var}\{\mathsf{z}_{ml}\,|\,\mathsf{x}_{nl}=x_{nl}\}=\nu^{a}_{l,mn}(t)x^{2}_{nl}+\nu^{p}_{n,ml}(t) \tag{4.16}$$

for

$$\widehat{p}_{n,ml}(t)\triangleq\sum_{k\neq n}\widehat{a}_{l,mk}(t)\widehat{x}_{m,kl}(t) \tag{4.17}$$

$$\begin{aligned}
\nu^{p}_{n,ml}(t)\triangleq&\sum_{k\neq n}\big(\widehat{a}^{2}_{l,mk}(t)\nu^{x}_{m,kl}(t)+\nu^{a}_{l,mk}(t)\widehat{x}^{2}_{m,kl}(t) \\
&+\nu^{a}_{l,mk}(t)\nu^{x}_{m,kl}(t)\big).
\end{aligned} \tag{4.18}$$

With this conditional-Gaussian approximation, (4.13) becomes

$$\Delta^{\mathsf{x}}_{m\to nl}(t,x_{nl}) \approx \mathsf{const} + \log \int_{z_{ml}} p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml}\,|\,z_{ml}) \tag{4.19}$$

$$\times\, \mathcal{N}\big(z_{ml}; \widehat{a}_{l,mn}(t)x_{nl}+\widehat{p}_{n,ml}(t), \nu^a_{l,mn}(t)x^2_{nl}+\nu^p_{n,ml}(t)\big)$$

$$= H_{ml}\Big(\widehat{a}_{l,mn}(t)x_{nl} + \widehat{p}_{n,ml}(t),$$

$$\nu^a_{l,mn}(t)x^2_{nl} + \nu^p_{n,ml}(t); y_{ml}\Big) + \mathsf{const} \tag{4.20}$$

in terms of the function

$$H_{ml}\big(\widehat{q},\nu^q;y\big) \triangleq \log \int_z p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y\,|\,z)\,\mathcal{N}(z;\widehat{q},\nu^q). \tag{4.21}$$

Unlike the original SPA message (4.7), the approximation (4.20) requires only a single integration. Still, additional simplifications are possible. First, notice that $\widehat{p}_{n,ml}(t)$ and $\nu^p_{n,ml}(t)$ differ from the corresponding $n$-invariant quantities

$$\widehat{p}_{ml}(t) \triangleq \sum_{k=1}^{N} \widehat{a}_{l,mk}(t)\widehat{x}_{m,kl}(t) \tag{4.22}$$

$$\nu^p_{ml}(t) \triangleq \sum_{k=1}^{N} \big(\widehat{a}^2_{l,mk}(t)\nu^x_{m,kl}(t) + \nu^a_{l,mk}(t)\widehat{x}^2_{m,kl}(t)$$

$$+ \nu^a_{l,mk}(t)\nu^x_{m,kl}(t)\big) \tag{4.23}$$

by one term. In the sequel, we will assume that $\widehat{p}_{ml}(t)$ and $\nu^p_{ml}(t)$ are $O(1)$ since these quantities can be recognized as the mean and variance, respectively, of an estimate

of $\mathsf{z}_{ml}$, which is $O(1)$. Writing the $H_{ml}$ term in (4.20) using (4.22)-(4.23),

$$\begin{aligned}
&H_{ml}\Big(\widehat{a}_{l,mn}(t)x_{nl}+\widehat{p}_{n,ml}(t),\nu^a_{l,mn}(t)x^2_{nl}+\nu^p_{n,ml}(t);y_{ml}\Big)\\
&=H_{ml}\Big(\widehat{a}_{l,mn}(t)\big(x_{nl}-\widehat{x}_{m,nl}(t)\big)+\widehat{p}_{ml}(t),\\
&\qquad\nu^a_{l,mn}(t)\big(x^2_{nl}-\widehat{x}^2_{m,nl}(t)\big)-\widehat{a}^2_{l,mn}(t)\nu^x_{m,nl}(t)\\
&\qquad-\nu^a_{l,mn}(t)\nu^x_{m,nl}(t)+\nu^p_{ml}(t);y_{ml}\Big) \qquad (4.24)\\
&=H_{ml}\Big(\widehat{a}_{l,mn}(t)\big(x_{nl}-\widehat{x}_{nl}(t)\big)+\widehat{p}_{ml}(t)+O(1/N),\\
&\qquad\nu^a_{l,mn}(t)\big(x^2_{nl}-\widehat{x}^2_{nl}(t)\big)+\nu^p_{ml}(t)+O(1/N);y_{ml}\Big) \qquad (4.25)
\end{aligned}$$

where in (4.25) we used the facts that $\widehat{a}_{l,mn}(t)(\widehat{x}_{nl}(t)-\widehat{x}_{m,nl}(t))$ and $\nu^a_{l,mn}(t)(\widehat{x}^2_{m,nl}(t)-\widehat{x}^2_{nl}(t)))-\widehat{a}^2_{l,mn}(t)\nu^x_{m,nl}(t)-\nu^a_{l,mn}(t)\nu^x_{m,nl}(t)$ are both $O(1/N)$.

Rewriting (4.20) using a Taylor series expansion in $x_{nl}$ about the point $\widehat{x}_{nl}(t)$, we get

$$\begin{aligned}
&\Delta^{\mathsf{x}}_{m\to nl}(t,x_{nl})\approx\mathsf{const}\\
&\quad+H_{ml}\big(\widehat{p}_{ml}(t)+O(1/N),\nu^p_{ml}(t)+O(1/N);y_{ml}\big)\\
&\quad+\widehat{a}_{l,mn}(t)\big(x_{nl}-\widehat{x}_{nl}(t)\big)\\
&\qquad\times H'_{ml}\big(\widehat{p}_{ml}(t)+O(1/N),\nu^p_{ml}(t)+O(1/N);y_{ml}\big)\\
&\quad+2\nu^a_{l,mn}(t)\widehat{x}_{nl}(t)\big(x_{nl}-\widehat{x}_{nl}(t)\big)\\
&\qquad\times\dot{H}_{ml}\big(\widehat{p}_{ml}(t)+O(1/N),\nu^p_{ml}(t)+O(1/N);y_{ml}\big)\\
&\quad+\nu^a_{l,mn}(t)\big(x_{nl}-\widehat{x}_{nl}(t)\big)^2\\
&\qquad\times\dot{H}_{ml}\big(\widehat{p}_{ml}(t)+O(1/N),\nu^p_{ml}(t)+O(1/N);y_{ml}\big)\\
&\quad+\frac{1}{2}\widehat{a}^2_{l,mn}(t)\big(x_{nl}-\widehat{x}_{nl}(t)\big)^2\\
&\qquad\times H''_{ml}\big(\widehat{p}_{ml}(t)+O(1/N),\nu^p_{ml}(t)+O(1/N);y_{ml}\big)\\
&\quad+O(1/N^{3/2}), \qquad (4.26)
\end{aligned}$$

where $H'_{ml}$ and $H''_{mn}$ are the first two derivatives of $H_{mn}$ w.r.t its first argument and $\dot{H}_{ml}$ is the first derivative w.r.t its second argument. Note that, in (4.26) and elsewhere, the higher-order terms in the Taylor's expansion are written solely in terms of their scaling dependence on $N$, which is what will eventually allow us to neglect these terms (in the large-system limit).

We now approximate (4.26) by dropping terms that vanish, relative to the second-to-last term in (4.26), as $N \to \infty$. Since this second-to-last term is $O(1/N)$ due to the scalings of $\widehat{a}^2_{l,mn}(t)$, $\widehat{p}_{ml}(t)$, and $\nu^p_{ml}(t)$, we drop terms that are of order $O(1/N^{3/2})$, such as the final term. We also replace $\nu^a_{l,mn}(t)$ with $\nu^a_{mn}(t)$, and $\widehat{a}^2_{l,mn}(t)$ with $\widehat{a}^2_{mn}(t)$, since in both cases the difference is $O(1/N^{3/2})$. Finally, we drop the $O(1/N)$ terms inside the $H_{ml}$ derivatives, which can be justified by taking a Taylor series expansion of these derivatives with respect to the $O(1/N)$ perturbations and verifying that the higher-order terms in this latter expansion are $O(1/N^{3/2})$. All of these approximations are analogous to those made in previous AMP derivations, e.g., [4], [16], and [6].

Applying these approximations to (4.26) and absorbing $x_{nl}$-invariant terms into the $\mathsf{const}$ term, we obtain

$$\begin{aligned}\Delta^{\mathsf{x}}_{m\to nl}(t, x_{nl}) \approx \big[\widehat{s}_{ml}(t)\widehat{a}_{l,mn}(t) + \nu^s_{ml}(t)\widehat{a}^2_{mn}(t)\widehat{x}_{nl}(t)\big] \\ \times x_{nl} - \tfrac{1}{2}\big[\nu^s_{ml}(t)\widehat{a}^2_{mn}(t) - \nu^a_{mn}(t) \\ \times(\widehat{s}^2_{ml}(t) - \nu^s_{ml}(t))\big]x^2_{nl} + \mathsf{const},\end{aligned} \tag{4.27}$$

where we used the relationship

$$\dot{H}_{ml}\big(\widehat{q}, \nu^q; y\big) \;=\; \frac{1}{2}\Big[\big(H'_{ml}\big(\widehat{q}, \nu^q; y\big)\big)^2 + H''_{ml}\big(\widehat{q}, \nu^q; y\big)\Big] \tag{4.28}$$

and defined

$$\widehat{s}_{ml}(t) \triangleq H'_{ml}\big(\widehat{p}_{ml}(t), \nu^p_{ml}(t); y_{ml}\big) \tag{4.29}$$

$$\nu^s_{ml}(t) \triangleq -H''_{ml}\big(\widehat{p}_{ml}(t), \nu^p_{ml}(t); y_{ml}\big). \tag{4.30}$$

Note that (4.27) is essentially a Gaussian approximation to the pdf $\frac{1}{C}\exp(\Delta^{\mathsf{x}}_{m\to nl}(t,.))$.

It was shown in Section 2.4 that

$$\widehat{s}_{ml}(t) = \frac{1}{\nu^p_{ml}(t)}\big(\widehat{z}_{ml}(t) - \widehat{p}_{ml}(t)\big) \tag{4.31}$$

$$\nu^s_{ml}(t) = \frac{1}{\nu^p_{ml}(t)}\left(1 - \frac{\nu^z_{ml}(t)}{\nu^p_{ml}(t)}\right), \tag{4.32}$$

for the conditional mean and variance

$$\widehat{z}_{ml}(t) \triangleq \mathrm{E}\{\mathsf{z}_{ml} \,|\, \mathsf{p}_{ml} = \widehat{p}_{ml}(t); \nu^p_{ml}(t)\} \tag{4.33}$$

$$\nu^z_{ml}(t) \triangleq \mathrm{var}\{\mathsf{z}_{ml} \,|\, \mathsf{p}_{ml} = \widehat{p}_{ml}(t); \nu^p_{ml}(t)\}, \tag{4.34}$$

computed according to the (conditional) pdf

$$\begin{aligned} &p_{\mathsf{z}_{ml}|\mathsf{p}_{ml}}\big(z_{ml} \,|\, \widehat{p}_{ml}(t); \nu^p_{ml}(t)\big) \\ &\quad\triangleq \tfrac{1}{C}\, p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml} \,|\, z_{ml})\, \mathcal{N}\big(z_{ml}; \widehat{p}_{ml}(t), \nu^p_{ml}(t)\big), \end{aligned} \tag{4.35}$$

where here $C = \int_z p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml} \,|\, z)\mathcal{N}\big(z; \widehat{p}_{ml}(t), \nu^p_{ml}(t)\big)$. In fact, (4.35) is BiG-AMP's iteration-$t$ approximation to the true marginal posterior $p_{\mathsf{z}_{ml}|\mathbf{Y}}(\cdot|\boldsymbol{Y})$. We note that (4.35) can also be interpreted as the (exact) posterior pdf for $\mathsf{z}_{ml}$ given the likelihood $p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml}|\cdot)$ from (4.3) and the prior $\mathsf{z}_{ml} \sim \mathcal{N}\big(\widehat{p}_{ml}(t), \nu^p_{ml}(t)\big)$ that is implicitly assumed by iteration-$t$ BiG-AMP.

Since $\mathbf{Z}^{\mathsf{T}} = \mathbf{X}^{\mathsf{T}}\mathbf{A}^{\mathsf{T}}$, the derivation of the BiG-AMP approximation of $\Delta^{\mathsf{a}}_{l\to mn}(t,.)$ closely follows the derivation for $\Delta^{\mathsf{x}}_{m\to nl}(t,.)$. In particular, it starts with (similar to

(4.13))

$$\begin{aligned}
&\Delta^{\mathsf{a}}_{l\to mn}(t, a_{mn}) \\
&= \log \int_{\{a_{mk}\}_{k\neq n}, \{x_{rl}\}_{r=1}^{N}} p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}\Big(y_{ml} \,\Big|\, \overbrace{a_{mn}x_{nl} + \sum_{k\neq n} a_{mk}x_{kl}}^{z_{ml}}\Big) \\
&\quad \times \prod_{r=1}^{N} \exp\Big(\Delta^{\mathsf{x}}_{m\leftarrow rl}(t, x_{rl})\Big) \prod_{k\neq n} \exp\Big(\Delta^{\mathsf{a}}_{l\leftarrow mk}(t, a_{mk})\Big) \\
&\quad + \mathsf{const}, \qquad (4.36)
\end{aligned}$$

where again the CLT motivates the treatment of $\mathsf{z}_{ml}$, conditioned on $\mathsf{a}_{mn} = a_{mn}$, as Gaussian. Eventually we arrive at the Taylor-series approximation (similar to (4.27))

$$\begin{aligned}
\Delta^{\mathsf{a}}_{l\to mn}(t, a_{mn}) \approx &\Big[\widehat{s}_{ml}(t)\widehat{x}_{m,nl}(t) + \nu^{s}_{ml}(t)\widehat{x}^2_{nl}(t)\widehat{a}_{mn}(t)\Big] \\
&\times a_{mn} - \tfrac{1}{2}\big[\nu^{s}_{ml}(t)\widehat{x}^2_{nl}(t) - \nu^{x}_{nl}(t) \\
&\times (\widehat{s}^2_{ml}(t) - \nu^{s}_{ml}(t))\big] a^2_{mn} \\
&+ \mathsf{const}. \qquad (4.37)
\end{aligned}$$

### 4.2.5 Approximated Variable-to-Factor Messages

We now turn to approximating the messages flowing from the variable nodes to the factor nodes. Starting with (4.8) and plugging in (4.27) we obtain

$$\begin{aligned}
&\Delta^{\mathsf{x}}_{m\leftarrow nl}(t+1, x_{nl}) \\
&\approx \mathsf{const} + \log p_{\mathsf{x}_{nl}}(x_{nl}) + \sum_{k\neq m} \Bigg( \Big[\widehat{s}_{kl}(t)\widehat{a}_{l,kn}(t) \\
&\quad + \nu^{s}_{kl}(t)\widehat{a}^2_{kn}(t)\widehat{x}_{nl}(t)\Big] x_{nl} - \tfrac{1}{2}\Big[\nu^{s}_{kl}(t)\widehat{a}^2_{kn}(t) \\
&\quad - \nu^{a}_{kn}(t)\big(\widehat{s}^2_{kl}(t) - \nu^{s}_{kl}(t)\big)\Big] x^2_{nl} \Bigg) \qquad (4.38) \\
&= \mathsf{const} + \log p_{\mathsf{x}_{nl}}(x_{nl}) - \frac{1}{2\nu^{r}_{m,nl}(t)}\big(x_{nl} - \widehat{r}_{m,nl}(t)\big)^2 \qquad (4.39) \\
&= \mathsf{const} + \log\Big(p_{\mathsf{x}_{nl}}(x_{nl})\mathcal{N}\big(x_{nl}; \widehat{r}_{m,nl}(t), \nu^{r}_{m,nl}(t)\big)\Big) \qquad (4.40)
\end{aligned}$$

for

$$\nu^r_{m,nl}(t) \triangleq \Bigg( \sum_{k\neq m} \widehat{a}^2_{kn}(t)\nu^s_{kl}(t) - \nu^a_{kn}(t)\big(\widehat{s}^2_{kl}(t) - \nu^s_{kl}(t)\big)\Bigg)^{-1} \tag{4.41}$$

$$\begin{aligned}\widehat{r}_{m,nl}(t) \triangleq \widehat{x}_{nl}(t)\Big(1 + \nu^r_{m,nl}(t) \sum_{k\neq m} \nu^a_{kn}(t)[\widehat{s}^2_{kl}(t) - \nu^s_{kl}(t)]\Big) \\ + \nu^r_{m,nl}(t) \sum_{k\neq m} \widehat{a}_{l,kn}(t)\widehat{s}_{kl}(t).\end{aligned} \tag{4.42}$$

Since $\widehat{a}^2_{mn}(t)$ and $\nu^a_{mn}(t)$ are $O(1/N)$, and recalling $\widehat{s}^2_{ml}(t)$ and $\nu^s_{ml}(t)$ are $O(1)$, we take $\nu^r_{m,nl}(t)$ to be $O(1)$. Meanwhile, since $\widehat{r}_{m,nl}(t)$ is an estimate of $\mathsf{x}_{nl}$, we reason that it is $O(1)$.

The mean and variance of the pdf associated with the $\Delta^{\mathsf{x}}_{m\leftarrow nl}(t{+}1,.)$ approximation in (4.40) are

$$\begin{aligned}&\widehat{x}_{m,nl}(t{+}1) \\ &\triangleq \underbrace{\frac{1}{C}\int_x x\, p_{\mathsf{x}_{nl}}(x)\mathcal{N}\big(x;\widehat{r}_{m,nl}(t),\nu^r_{m,nl}(t)\big)}_{\triangleq\, g_{\mathsf{x}_{nl}}(\widehat{r}_{m,nl}(t),\nu^r_{m,nl}(t))}\end{aligned} \tag{4.43}$$

$$\begin{aligned}&\nu^x_{m,nl}(t{+}1) \\ &\triangleq \underbrace{\frac{1}{C}\int_x \big|x - \widehat{x}_{m,nl}(t{+}1)\big|^2 p_{\mathsf{x}_{nl}}(x)\mathcal{N}\big(x;\widehat{r}_{m,nl}(t),\nu^r_{m,nl}(t)\big)}_{\nu^r_{m,nl}(t)\, g'_{\mathsf{x}_{nl}}(\widehat{r}_{m,nl}(t),\nu^r_{m,nl}(t))}\end{aligned} \tag{4.44}$$

where here $C = \int_x p_{\mathsf{x}_{nl}}(x)\mathcal{N}\big(x;\widehat{r}_{m,nl}(t),\nu^r_{m,nl}(t)\big)$ and $g'_{\mathsf{x}_n l}$ denotes the derivative of $g_{\mathsf{x}_{nl}}$ with respect to the first argument. The fact that (4.43) and (4.44) are related through a derivative was shown in [6].

We now derive approximations of $\widehat{x}_{m,nl}(t)$ and $\nu^x_{m,nl}(t)$ that avoid the dependence on the destination node $m$. For this, we introduce $m$-invariant versions of $\widehat{r}_{m,nl}(t)$

and $\nu^r_{m,nl}(t)$:

$$\nu^r_{nl}(t) \triangleq \left( \sum_{m=1}^{M} \widehat{a}^2_{mn}(t)\nu^s_{ml}(t) - \nu^a_{mn}(t)\big(\widehat{s}^2_{ml}(t) - \nu^s_{ml}(t)\big) \right)^{-1} \tag{4.45}$$

$$\begin{aligned}\widehat{r}_{nl}(t) \triangleq\ & \widehat{x}_{nl}(t)\Big(1 + \nu^r_{nl}(t) \sum_{m=1}^{M} \nu^a_{mn}(t)[\widehat{s}^2_{ml}(t) - \nu^s_{ml}(t)]\Big) \\ & + \nu^r_{nl}(t) \sum_{m=1}^{M} \widehat{a}_{l,mn}(t)\widehat{s}_{ml}(t).\end{aligned} \tag{4.46}$$

Comparing (4.45)-(4.46) with (4.41)-(4.42) and applying previously established scalings from Table 4.2 reveals that $\nu^r_{m,nl}(t) - \nu^r_{nl}(t)$ is $O(1/N)$ and that $\widehat{r}_{m,nl}(t) = \widehat{r}_{nl}(t) - \nu^r_{nl}(t)\widehat{a}_{mn}(t)\widehat{s}_{ml}(t) + O(1/N)$, so that (4.43) implies

$$\begin{aligned}&\widehat{x}_{m,nl}(t+1) \\ &= g_{\mathsf{x}_{nl}}\big(\widehat{r}_{nl}(t) - \nu^r_{nl}(t)\widehat{a}_{mn}(t)\widehat{s}_{ml}(t) + O(1/N), \\ &\qquad \nu^r_{nl}(t) + O(1/N)\big) && (4.47) \\ &= g_{\mathsf{x}_{nl}}\big(\widehat{r}_{nl}(t) - \nu^r_{nl}(t)\widehat{a}_{mn}(t)\widehat{s}_{ml}(t), \nu^r_{nl}(t)\big) + O(1/N) && (4.48) \\ &= g_{\mathsf{x}_{nl}}\big(\widehat{r}_{nl}(t), \nu^r_{nl}(t)\big) && (4.49) \\ &\qquad -\nu^r_{nl}(t)\widehat{a}_{mn}(t)\widehat{s}_{ml}(t)\, g'_{\mathsf{x}_{nl}}\big(\widehat{r}_{nl}(t), \nu^r_{nl}(t)\big) + O(1/N) \\ &\approx \widehat{x}_{nl}(t+1) - \widehat{a}_{mn}(t)\widehat{s}_{ml}(t)\nu^x_{nl}(t+1). && (4.50)\end{aligned}$$

Above, (4.48) follows from taking Taylor series expansions around each of the $O(1/N)$ perturbations in (4.47); (4.49) follows from a Taylor series expansion in the first argument of (4.48) about the point $\widehat{r}_{nl}(t)$; and (4.50) follows by neglecting the $O(1/N)$ term (which vanishes relative to the others in the large-system limit) and applying the definitions

$$\widehat{x}_{nl}(t+1) \triangleq g_{\mathsf{x}_{nl}}\big(\widehat{r}_{nl}(t), \nu^r_{nl}(t)\big) \tag{4.51}$$

$$\nu^x_{nl}(t+1) \triangleq \nu^r_{nl}(t) g'_{\mathsf{x}_{nl}}\big(\widehat{r}_{nl}(t), \nu^r_{nl}(t)\big), \tag{4.52}$$

which match (4.43)-(4.44) sans the $m$ dependence. Note that (4.50) confirms that the difference $\widehat{x}_{m,nl}(t) - \widehat{x}_{nl}(t)$ is $O(1/\sqrt{N})$, as was assumed at the start of the BiG-AMP derivation. Likewise, taking Taylor series expansions of $g'_{\mathsf{x}_{nl}}$ in (4.44) about the point $\widehat{r}_{nl}(t)$ in the first argument and about the point $\nu^r_{nl}(t)$ in the second argument and then comparing the result with (4.52) confirms that $\nu^x_{m,nl}(t) - \nu^x_{nl}(t)$ is $O(1/\sqrt{N})$.

We then repeat the above procedure to derive an approximation to $\Delta^{\mathsf{a}}_{l\leftarrow mn}(t+1,.)$ analogous to (4.40), whose corresponding mean is then further approximated as

$$\widehat{a}_{l,mn}(t+1) \approx \widehat{a}_{mn}(t+1) - \widehat{x}_{nl}(t)\widehat{s}_{ml}(t)\nu^a_{mn}(t+1), \tag{4.53}$$

for

$$\widehat{a}_{mn}(t+1) \triangleq g_{\mathsf{a}_{mn}}\big(\widehat{q}_{mn}(t), \nu^q_{mn}(t)\big) \tag{4.54}$$

$$\nu^a_{mn}(t+1) \triangleq \nu^q_{mn}(t) g'_{\mathsf{a}_{mn}}\big(\widehat{q}_{mn}(t), \nu^q_{mn}(t)\big) \tag{4.55}$$

$$g_{\mathsf{a}_{mn}}(\widehat{q}, \nu^q) \triangleq \frac{\int_a a\, p_{\mathsf{a}_{mn}}(a)\mathcal{N}(a;\widehat{q},\nu^q)}{\int_a p_{\mathsf{a}_{mn}}(a)\mathcal{N}(a;\widehat{q},\nu^q)} \tag{4.56}$$

where

$$\nu^q_{mn}(t) \triangleq \bigg(\sum_{l=1}^{L} \widehat{x}^2_{nl}(t)\nu^s_{ml}(t) - \nu^x_{nl}(t)\big(\widehat{s}^2_{ml}(t) - \nu^s_{ml}(t)\big)\bigg)^{-1} \tag{4.57}$$

$$\begin{aligned}\widehat{q}_{mn}(t) \triangleq \widehat{a}_{mn}(t)\Big(1 + \nu^q_{mn}(t)\sum_{l=1}^{L}\nu^x_{nl}(t)[\widehat{s}^2_{ml}(t) - \nu^s_{ml}(t)]\Big) \\ + \nu^q_{mn}(t)\sum_{l=1}^{L}\widehat{x}_{m,nl}(t)\widehat{s}_{ml}(t).\end{aligned} \tag{4.58}$$

Arguments analogous to the discussion following (4.42) justify the remaining scalings in Table 4.2.

### 4.2.6 Closing the Loop

The penultimate step in the derivation of BiG-AMP is to approximate earlier steps that use $\widehat{a}_{l,mn}(t)$ and $\widehat{x}_{m,nl}(t)$ in place of $\widehat{a}_{mn}(t)$ and $\widehat{x}_{nl}(t)$. For this, we start

by plugging (4.50) and (4.53) into (4.22), which yields[13]

$$\begin{aligned}
\widehat{p}_{ml}(t) \;=\; & O(1/\sqrt{N}) + \overbrace{\sum_{n=1}^{N} \widehat{a}_{mn}(t)\widehat{x}_{nl}(t)}^{\triangleq\, \overline{p}_{ml}(t)} - \widehat{s}_{ml}(t-1) \\
& \times \sum_{n=1}^{N} \Big( \nu^a_{mn}(t)\widehat{x}_{nl}(t)\widehat{x}_{nl}(t-1) + \widehat{a}_{mn}(t)\widehat{a}_{mn}(t-1)\nu^x_{nl}(t) \Big) \\
& + \widehat{s}^2_{ml}(t-1) \sum_{n=1}^{N} \widehat{a}_{mn}(t-1)\nu^a_{mn}(t)\nu^x_{nl}(t)\widehat{x}_{nl}(t-1) \qquad (4.59) \\
\approx\; & \overline{p}_{ml}(t) - \widehat{s}_{ml}(t-1) \underbrace{\sum_{n=1}^{N} \Big( \nu^a_{mn}(t)\widehat{x}^2_{nl}(t) + \widehat{a}^2_{mn}(t)\nu^x_{nl}(t) \Big)}_{\triangleq\, \overline{\nu}^p_{ml}(t)}, \qquad (4.60)
\end{aligned}$$

where, for (4.60), we used $\widehat{a}^2_{mn}(t)$ in place of $\widehat{a}_{mn}(t)\widehat{a}_{mn}(t-1)$, used $\widehat{x}^2_{nl}(t)$ in place of $\widehat{x}_{nl}(t)\widehat{x}_{nl}(t-1)$, and neglected terms that are $O(1/\sqrt{N})$, since they vanish relative to the remaining $O(1)$ terms in the large-system limit.

Next we plug (4.50), (4.53), $\nu^x_{m,nl}(t) = \nu^x_{nl}(t) + O(1/\sqrt{N})$, and $\nu^a_{l,mn}(t) = \nu^a_{mn}(t) + O(1/N^{3/2})$ into (4.23), giving

$$\begin{aligned}
\nu^p_{ml}(t) = \overline{\nu}^p_{ml}(t) & + \sum_{n=1}^{N} \nu^a_{mn}(t)\nu^x_{nl}(t) \qquad (4.61) \\
& - 2\widehat{s}_{ml}(t-1) \sum_{n=1}^{N} \Big( \nu^a_{mn}(t)\widehat{a}_{mn}(t)\widehat{x}_{nl}(t-1)\nu^x_{nl}(t) \\
& \quad + \nu^a_{mn}(t)\widehat{a}_{mn}(t-1)\widehat{x}_{nl}(t)\nu^x_{nl}(t) \Big) \\
& + \widehat{s}_{ml}(t-1)^2 \sum_{n=1}^{N} \Big( (\nu^a_{mn}(t))^2 \widehat{x}^2_{nl}(t-1)\nu^x_{nl}(t) \\
& \quad + \nu^a_{mn}(t)\widehat{a}^2_{mn}(t-1)(\nu^x_{nl}(t))^2 \Big) + O(1/\sqrt{N}) \\
\approx\; & \overline{\nu}^p_{ml}(t) + \sum_{n=1}^{N} \nu^a_{mn}(t)\nu^x_{nl}(t), \qquad (4.62)
\end{aligned}$$

[13]Recall that the error of the approximation in (4.50) is $O(1/N)$ and the error in (4.53) is $O(1/N^{3/2})$.

where (4.62) retains only the $O(1)$ terms from (4.61).

Similarly, we plug (4.53) into (4.46) and (4.50) into (4.58) to obtain

$$\widehat{r}_{nl}(t) \approx \widehat{x}_{nl}(t)\left(1 - \frac{\sum_{m=1}^{M}\nu^a_{mn}(t)\nu^s_{ml}(t)}{\sum_{m=1}^{M}\widehat{a}^2_{mn}(t)\nu^s_{ml}(t)}\right) + \nu^r_{nl}(t)\sum_{m=1}^{M}\widehat{a}_{mn}(t)\widehat{s}_{ml}(t) \tag{4.63}$$

$$\widehat{q}_{mn}(t) \approx \widehat{a}_{mn}(t)\left(1 - \frac{\sum_{l=1}^{L}\nu^x_{nl}(t)\nu^s_{ml}(t)}{\sum_{l=1}^{L}\widehat{x}^2_{nl}(t)\nu^s_{ml}(t)}\right) + \nu^q_{mn}(t)\sum_{l=1}^{L}\widehat{x}_{nl}(t)\widehat{s}_{ml}(t), \tag{4.64}$$

where the approximations involve the use of $\widehat{s}^2_{ml}(t)$ in place of $\widehat{s}_{ml}(t)\widehat{s}_{ml}(t-1)$, of $\widehat{a}_{mn}(t)$ in place of $\widehat{a}_{mn}(t-1)$, of $\widehat{x}_{nl}(t)$ in place of $\widehat{x}_{nl}(t-1)$, and the dropping of terms that vanish in the large-system limit. Finally, we make the approximations

$$\nu^r_{nl}(t) \approx \left(\sum_{m=1}^{M}\widehat{a}^2_{mn}(t)\nu^s_{ml}(t)\right)^{-1} \tag{4.65}$$

$$\nu^q_{mn}(t) \approx \left(\sum_{l=1}^{L}\widehat{x}^2_{nl}(t)\nu^s_{ml}(t)\right)^{-1}, \tag{4.66}$$

by neglecting the $\widehat{s}^2_{ml}(t) - \nu^s_{ml}(t)$ terms in (4.45) and (4.57).

Here we pause to explain the approximations (4.65)-(4.66). The term neglected in going from (4.45) to (4.65) can be written using (4.31)-(4.32) as

$$\begin{aligned}
&\sum_{m=1}^{M}\nu^a_{mn}(t)\big(\widehat{s}^2_{ml}(t) - \nu^s_{ml}(t)\big) \\
&= \sum_{m=1}^{M}\nu^a_{mn}(t)\left[\frac{(\widehat{z}_{ml}(t) - \widehat{p}_{ml}(t))^2 + \nu^z_{ml}(t)}{\nu^p_{ml}(t)^2} - \frac{1}{\nu^p_{ml}(t)}\right] &(4.67)\\
&= \sum_{m=1}^{M}\frac{\nu^a_{mn}(t)}{\nu^p_{ml}(t)}\left[\mathrm{E}\left\{\frac{(\mathsf{z}_{ml} - \widehat{p}_{ml}(t))^2}{\nu^p_{ml}(t)}\right\} - 1\right] &(4.68)
\end{aligned}$$

where the expectations are taken over $\mathsf{z}_{ml} \sim p_{\mathsf{z}_{ml}|\mathsf{p}_{ml}}\big(\,\cdot\,|\,\widehat{p}_{ml}(t); \nu^p_{ml}(t)\big)$ from (4.35). For GAMP, [6, Sec. VI.D] clarifies that, in the large system limit, under i.i.d priors

and scalar variances, the true $z_m$ and the iterates $\widehat{p}_m(t)$ converge empirically to a pair of random variables $(\mathsf{z}, \mathsf{p})$ that satisfy $p_{\mathsf{z}\,|\,\mathsf{p}}(z\,|\,\widehat{p}(t)) = \mathcal{N}(z; \widehat{p}(t), \nu^p(t))$. This result leads us to believe that the expectation in (4.68) is approximately unit-valued when averaged over $m$, and thus (4.68) is approximately zero-valued. Similar reasoning applies to (4.66).

### 4.2.7 Approximated Posteriors

The final step in the BiG-AMP derivation is to approximate the SPA posterior log-pdfs in (4.11) and (4.12). Plugging (4.27) and (4.37) into those expressions, we get

$$\begin{aligned}
&\Delta^{\mathsf{x}}_{nl}(t+1, x_{nl}) \\
&\approx \mathsf{const} + \log\Big(p_{\mathsf{x}_{nl}}(x_{nl})\,\mathcal{N}\big(x_{nl}; \widehat{r}_{nl}(t), \nu^r_{nl}(t)\big)\Big) \qquad (4.69) \\
&\Delta^{\mathsf{a}}_{mn}(t+1, a_{mn}) \\
&\approx \mathsf{const} + \log\Big(p_{\mathsf{a}_{mn}}(a_{mn})\,\mathcal{N}\big(a_{mn}; \widehat{q}_{mn}(t), \nu^q_{mn}(t)\big)\Big) \qquad (4.70)
\end{aligned}$$

using steps similar to (4.40). The associated pdfs are

$$\begin{aligned}
&p_{\mathsf{x}_{nl}|\mathsf{r}_{nl}}\big(x_{nl}\,|\,\widehat{r}_{nl}(t); \nu^r_{nl}(t)\big) \\
&\triangleq \tfrac{1}{C_x}\, p_{\mathsf{x}_{nl}}(x_{nl})\,\mathcal{N}\big(x_{nl}; \widehat{r}_{nl}(t), \nu^r_{nl}(t)\big) \qquad (4.71) \\
&p_{\mathsf{a}_{mn}|\mathsf{q}_{mn}}\big(a_{mn}\,|\,\widehat{q}_{mn}(t); \nu^q_{mn}(t)\big) \\
&\triangleq \tfrac{1}{C_a}\, p_{\mathsf{a}_{mn}}(a_{mn})\,\mathcal{N}\big(a_{mn}; \widehat{q}_{mn}(t), \nu^q_{mn}(t)\big) \qquad (4.72)
\end{aligned}$$

for $C_x \triangleq \int_x p_{\mathsf{x}_{nl}}(x)\mathcal{N}\big(x; \widehat{r}_{nl}(t), \nu^r_{nl}(t)\big)$ and $C_a \triangleq \int_a p_{\mathsf{a}_{mn}}(a)\mathcal{N}\big(a; \widehat{q}_{mn}(t), \nu^q_{mn}(t)\big)$, which are iteration-$t$ BiG-AMP's approximations to the true marginal posteriors $p_{\mathsf{x}_{nl}|\boldsymbol{\mathsf{Y}}}(x_{nl}\,|\,\boldsymbol{Y})$ and $p_{\mathsf{a}_{mn}|\boldsymbol{\mathsf{Y}}}(a_{mn}\,|\,\boldsymbol{Y})$, respectively.

Note that $\widehat{x}_{nl}(t+1)$ and $\nu^x_{nl}(t+1)$ from (4.51)-(4.52) are the mean and variance, respectively, of the posterior pdf in (4.71). Note also that (4.71) can be interpreted as the (exact) posterior pdf of $\mathsf{x}_{nl}$ given the observation $\mathsf{r}_{nl} = \widehat{r}_{nl}(t)$ under the prior model $\mathsf{x}_{nl} \sim p_{\mathsf{x}_{nl}}$ and the likelihood model $p_{\mathsf{r}_{nl}|\mathsf{x}_{nl}}(\widehat{r}_{nl}(t) \,|\, x_{nl}; \nu^r_{nl}(t)) = \mathcal{N}\big(\widehat{r}_{nl}(t); x_{nl}, \nu^r_{nl}(t)\big)$ implicitly assumed by iteration-$t$ BiG-AMP. Analogous statements can be made about the posterior pdf of $\mathsf{a}_{mn}$ in (4.72).

This completes the derivation of BiG-AMP.

### 4.2.8 Algorithm Summary

The BiG-AMP algorithm derived in Sections 4.2.3 to 4.2.7 is summarized in Table 4.3. There, we have included a maximum number of iterations, $T_{\max}$ and a stopping condition (R17) based on the (normalized) change in the residual and a user-defined parameter $\tau_{\text{BiG-AMP}}$. We have also written the algorithm in a more general form that allows the use of complex-valued quantities [note the complex conjugates in (R10) and (R12)], in which case $\mathcal{N}$ in (D1)-(D3) would be circular complex Gaussian. For ease of interpretation, Table 4.3 does not include the important damping modifications that will be detailed in Section 4.4.1. Suggestions for the initializations in (I2) will be given in the sequel.

We note that BiG-AMP avoids the use of SVD or QR decompositions, lending itself to simple and potentially parallel implementations. Its complexity order is dominated[14] by ten matrix multiplications per iteration [in steps (R1)-(R3) and (R9)-(R12)], each requiring $MNL$ multiplications, although simplifications will be discussed in Section 4.3.

[14]The computations in steps (R4)-(R8) are $O(ML)$, while the remainder of the algorithm is $O(MN + NL)$. Thus, as $N$ grows, the matrix multiplies dominate the complexity.

definitions:

$$p_{\mathsf{z}_{ml}|\mathsf{p}_{ml}}(z|\widehat{p};\nu^p) \triangleq \frac{p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml}|z)\,\mathcal{N}(z;\widehat{p},\nu^p)}{\int_{z'} p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml}|z')\,\mathcal{N}(z';\widehat{p},\nu^p)} \quad \text{(D1)}$$

$$p_{\mathsf{x}_{nl}|\mathsf{r}_{nl}}(x|\widehat{r};\nu^r) \triangleq \frac{p_{\mathsf{x}_{nl}}(x)\,\mathcal{N}(x;\widehat{r},\nu^r)}{\int_{x'} p_{\mathsf{x}_{nl}}(x')\,\mathcal{N}(x';\widehat{r},\nu^r)} \quad \text{(D2)}$$

$$p_{\mathsf{a}_{mn}|\mathsf{q}_{mn}}(a|\widehat{q};\nu^q) \triangleq \frac{p_{\mathsf{a}_{mn}}(a)\,\mathcal{N}(a;\widehat{q},\nu^q)}{\int_{a'} p_{\mathsf{a}_{mn}}(a')\,\mathcal{N}(a';\widehat{q},\nu^q)} \quad \text{(D3)}$$

initialization:

$$\forall m,l: \widehat{s}_{ml}(0) = 0 \quad \text{(I1)}$$

$$\forall m,n,l: \text{choose } \nu^x_{nl}(1), \widehat{x}_{nl}(1), \nu^a_{mn}(1), \widehat{a}_{mn}(1) \quad \text{(I2)}$$

for $t = 1, \ldots T_{\max}$

$$\forall m,l: \overline{\nu}^p_{ml}(t) = \textstyle\sum_{n=1}^N |\widehat{a}_{mn}(t)|^2 \nu^x_{nl}(t) + \nu^a_{mn}(t)|\widehat{x}_{nl}(t)|^2 \quad \text{(R1)}$$

$$\forall m,l: \overline{p}_{ml}(t) = \textstyle\sum_{n=1}^N \widehat{a}_{mn}(t)\widehat{x}_{nl}(t) \quad \text{(R2)}$$

$$\forall m,l: \nu^p_{ml}(t) = \overline{\nu}^p_{ml}(t) + \textstyle\sum_{n=1}^N \nu^a_{mn}(t)\nu^x_{nl}(t) \quad \text{(R3)}$$

$$\forall m,l: \widehat{p}_{ml}(t) = \overline{p}_{ml}(t) - \widehat{s}_{ml}(t-1)\overline{\nu}^p_{ml}(t) \quad \text{(R4)}$$

$$\forall m,l: \nu^z_{ml}(t) = \mathrm{var}\{\mathsf{z}_{ml} \,|\, \mathsf{p}_{ml} = \widehat{p}_{ml}(t); \nu^p_{ml}(t)\} \quad \text{(R5)}$$

$$\forall m,l: \widehat{z}_{ml}(t) = \mathrm{E}\{\mathsf{z}_{ml} \,|\, \mathsf{p}_{ml} = \widehat{p}_{ml}(t); \nu^p_{ml}(t)\} \quad \text{(R6)}$$

$$\forall m,l: \nu^s_{ml}(t) = (1 - \nu^z_{ml}(t)/\nu^p_{ml}(t))/\nu^p_{ml}(t) \quad \text{(R7)}$$

$$\forall m,l: \widehat{s}_{ml}(t) = (\widehat{z}_{ml}(t) - \widehat{p}_{ml}(t))/\nu^p_{ml}(t) \quad \text{(R8)}$$

$$\forall n,l: \nu^r_{nl}(t) = \big(\textstyle\sum_{m=1}^M |\widehat{a}_{mn}(t)|^2 \nu^s_{ml}(t)\big)^{-1} \quad \text{(R9)}$$

$$\forall n,l: \widehat{r}_{nl}(t) = \widehat{x}_{nl}(t)(1 - \nu^r_{nl}(t)\textstyle\sum_{m=1}^M \nu^a_{mn}(t)\nu^s_{ml}(t)) + \nu^r_{nl}(t)\textstyle\sum_{m=1}^M \widehat{a}^*_{mn}(t)\widehat{s}_{ml}(t) \quad \text{(R10)}$$

$$\forall m,n: \nu^q_{mn}(t) = \big(\textstyle\sum_{l=1}^L |\widehat{x}_{nl}(t)|^2 \nu^s_{ml}(t)\big)^{-1} \quad \text{(R11)}$$

$$\forall m,n: \widehat{q}_{mn}(t) = \widehat{a}_{mn}(t)(1 - \nu^q_{mn}(t)\textstyle\sum_{l=1}^L \nu^x_{nl}(t)\nu^s_{ml}(t)) + \nu^q_{mn}(t)\textstyle\sum_{l=1}^L \widehat{x}^*_{nl}(t)\widehat{s}_{ml}(t) \quad \text{(R12)}$$

$$\forall n,l: \nu^x_{nl}(t+1) = \mathrm{var}\{\mathsf{x}_{nl} \,|\, \mathsf{r}_{nl} = \widehat{r}_{nl}(t); \nu^r_{nl}(t)\} \quad \text{(R13)}$$

$$\forall n,l: \widehat{x}_{nl}(t+1) = \mathrm{E}\{\mathsf{x}_{nl} \,|\, \mathsf{r}_{nl} = \widehat{r}_{nl}(t); \nu^r_{nl}(t)\} \quad \text{(R14)}$$

$$\forall m,n: \nu^a_{mn}(t+1) = \mathrm{var}\{\mathsf{a}_{mn} \,|\, \mathsf{q}_{mn} = \widehat{q}_{mn}(t); \nu^q_{mn}(t)\} \quad \text{(R15)}$$

$$\forall m,n: \widehat{a}_{mn}(t+1) = \mathrm{E}\{\mathsf{a}_{mn} \,|\, \mathsf{q}_{mn} = \widehat{q}_{mn}(t); \nu^q_{mn}(t)\} \quad \text{(R16)}$$

$$\text{if } \textstyle\sum_{m,l} |\overline{p}_{ml}(t) - \overline{p}_{ml}(t-1)|^2 \le \tau_{\text{BiG-AMP}} \sum_{m,l} |\overline{p}_{ml}(t)|^2, \text{ stop} \quad \text{(R17)}$$

end

Table 4.3: The BiG-AMP Algorithm

The steps in Table 4.3 can be interpreted as follows. (R1)-(R2) compute a "plug-in" estimate $\overline{\boldsymbol{P}}$ of the matrix product $\mathbf{Z} = \mathbf{AX}$ and a corresponding set of element-wise variances $\{\overline{\nu}^p_{ml}\}$. (R3)-(R4) then apply "Onsager" correction (see [16] and [6] for discussions in the contexts of AMP and GAMP, respectively) to obtain the corresponding quantities $\widehat{\boldsymbol{P}}$ and $\{\nu^p_{ml}\}$. Using these quantities, (R5)-(R6) compute the (approximate) marginal posterior means $\widehat{\boldsymbol{Z}}$ and variances $\{\nu^z_{ml}\}$ of $\mathbf{Z}$. Steps (R7)-(R8) then use these posterior moments to compute the scaled residual $\widehat{\boldsymbol{S}}$ and a set of inverse-residual-variances $\{\nu^s_{ml}\}$. This interpretation becomes clear in the case of AWGN observations with noise variance $\nu^w$, where

$$p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml} \,|\, z_{ml}) = \mathcal{N}(y_{ml}; z_{ml}, \nu^w). \tag{4.73}$$

and hence

$$\nu^s_{ml} = \frac{1}{\nu^p_{ml} + \nu^w} \quad \text{and} \quad \widehat{s}_{ml} = \frac{y_{ml} - \widehat{p}_{ml}}{\nu^p_{ml} + \nu^w}. \tag{4.74}$$

Steps (R9)-(R10) then use the residual terms $\widehat{\boldsymbol{S}}$ and $\{\nu^s_{ml}\}$ to compute $\widehat{\boldsymbol{R}}$ and $\{\nu^r_{nl}\}$, where $\widehat{r}_{nl}$ can be interpreted as a $\nu^r_{nl}$-variance-AWGN corrupted observation of the true $\mathsf{x}_{nl}$. Similarly, (R11)-(R12) compute $\widehat{\boldsymbol{Q}}$ and $\{\nu^q_{mn}\}$, where $\widehat{q}_{mn}$ can be interpreted as a $\nu^q_{mn}$-variance-AWGN corrupted observation of the true $\mathsf{a}_{mn}$. Finally, (R13)-(R14) merge these AWGN-corrupted observations with the priors $\{p_{\mathsf{x}_{nl}}\}$ to produce the posterior means $\widehat{\boldsymbol{X}}$ and variances $\{\nu^x_{nl}\}$; (R15)-(R16) do the same for the $\mathsf{a}_{mn}$ quantities.

The BiG-AMP algorithm in Table 4.3 is a direct (although non-trivial) extension of the GAMP algorithm for compressive sensing [6] described in Chapter 2, which estimates $\mathbf{X}$ assuming perfectly known $\mathbf{A}$, and bears even stronger similarities to the

MU-GAMP [7] algorithm described in Chapter 3, which estimates $\mathbf{X}$ assuming knowledge of the marginal means and variances of unknown random $\mathbf{A}$, but which makes no attempt to estimate $\mathbf{A}$ itself. In Section 4.3.2, a simplified version of BiG-AMP will be developed that is similar to the Bayesian-AMP algorithm [4] for compressive sensing.

## 4.3 BiG-AMP Simplifications

We now describe simplifications of the BiG-AMP algorithm from Table 4.3 that result from additional approximations and from the use of specific priors $p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}$, $p_{\mathsf{x}_{nl}}$, and $p_{\mathsf{a}_{mn}}$ that arise in practical applications of interest.

### 4.3.1 Scalar Variances

The BiG-AMP algorithm in Table 4.3 stores and processes a number of element-wise variance terms whose values vary across the elements (e.g., $\nu^x_{nl}$ can vary across $n$ and $l$). The use of scalar variances (i.e., uniform across $m, n, l$) significantly reduces the memory and complexity of the algorithm.

To derive scalar-variance BiG-AMP, we first assume $\forall n, l : \nu^x_{nl}(t) \approx \nu^x(t) \triangleq \frac{1}{NL}\sum_{n=1}^{N}\sum_{l=1}^{L}\nu^x_{nl}(t)$ and $\forall m, n : \nu^a_{mn}(t) \approx \nu^a(t) \triangleq \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\nu^a_{mn}(t)$, so from (R1)

$$\overline{\nu}^p_{ml}(t) \approx \nu^x(t)\sum_{n=1}^{N}|\widehat{a}_{mn}(t)|^2 + \nu^a(t)\sum_{n=1}^{N}|\widehat{x}_{nl}(t)|^2 \tag{4.75}$$

$$\approx \frac{\|\widehat{\boldsymbol{A}}(t)\|_F^2}{M}\nu^x(t) + \frac{\|\widehat{\boldsymbol{X}}(t)\|_F^2}{L}\nu^a(t) \triangleq \overline{\nu}^p(t). \tag{4.76}$$

Note that using (4.76) in place of (R1) avoids two matrix multiplies. Plugging these approximations into (R3) gives

$$\nu^p_{ml}(t) \approx \overline{\nu}^p(t) + N\nu^a(t)\nu^x(t) \triangleq \nu^p(t) \tag{4.77}$$

which, when used in place of (R3), avoids another matrix multiply. Even with the above scalar-variance approximations, $\{\nu^s_{ml}(t)\}$ from (R5) are not guaranteed to be equal (except in special cases like AWGN $p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}$). Still, they can be approximated as such using $\nu^s(t) \triangleq \frac{1}{ML}\sum_{m=1}^{M}\sum_{l=1}^{L}\nu^s_{ml}(t)$, in which case

$$\nu^r_{nl}(t) \approx \frac{1}{\nu^s(t)\sum_{m=1}^{M}|\widehat{a}_{mn}(t)|^2} \approx \frac{N}{\nu^s(t)\|\widehat{\boldsymbol{A}}(t)\|_F^2} \triangleq \nu^r(t) \tag{4.78}$$

$$\nu^q_{mn}(t) \approx \frac{1}{\nu^s(t)\sum_{l=1}^{L}|\widehat{x}_{nl}(t)|^2} \approx \frac{N}{\nu^s(t)\|\widehat{\boldsymbol{X}}(t)\|_F^2} \triangleq \nu^q(t). \tag{4.79}$$

Using (4.78) in place of (R9) and (4.79) in place of (R11) avoids two matrix multiplies and $NL+MN-2$ scalar divisions, and furthermore allows (R10) and (R12) to be implemented as

$$\widehat{r}_{nl}(t) = \widehat{x}_{nl}(t)\left(1 - \frac{MN\nu^a(t)}{\|\widehat{\boldsymbol{A}}(t)\|_F^2}\right) + \nu^r(t)\sum_{m=1}^{M}\widehat{a}_{mn}(t)\widehat{s}_{ml}(t) \tag{4.80}$$

$$\widehat{q}_{mn}(t) = \widehat{a}_{mn}(t)\left(1 - \frac{NL\nu^x(t)}{\|\widehat{\boldsymbol{X}}(t)\|_F^2}\right) + \nu^q(t)\sum_{l=1}^{L}\widehat{x}_{nl}(t)\widehat{s}_{ml}(t), \tag{4.81}$$

saving two more matrix multiplies, and leaving a total of only three matrix multiplies per iteration.

### 4.3.2 Possibly Incomplete AWGN Observations

We now consider a particular observation model wherein the elements of $\boldsymbol{Z}=\boldsymbol{A}\boldsymbol{X}$ are AWGN-corrupted at a subset of indices $\Omega \subset (1\dots M)\times(1\dots L)$ and unobserved at the remaining indices, noting that the standard AWGN model (4.73) is the special

case where $|\Omega| = ML$. This "possibly incomplete AWGN" (PIAWGN) model arises in a number of important applications, such as matrix completion and dictionary learning.

We can state the PIAWGN model probabilistically as

$$p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml} \,|\, z_{ml}) = \begin{cases} \mathcal{N}(y_{ml}; z_{ml}, \nu^w) & (m,l) \in \Omega \\ 1_{y_{ml}} & (m,l) \notin \Omega, \end{cases} \tag{4.82}$$

where $\nu^w$ is the noise variance on the non-missing observations and $1_y$ denotes a point mass at $y=0$. Thus, at the observed entries $(m,l) \in \Omega$, the quantities $\widehat{s}_{ml}$ and $\nu^s_{ml}$ calculated using the AWGN expressions (4.74), while at the "missing" entries $(m,l) \notin \Omega$, where $y_{ml}$ is invariant to $z_{ml}$, we have $\mathrm{E}\{\mathsf{z}_{ml} \,|\, \mathsf{p}_{ml} = \widehat{p}_{ml}; \nu^p_{ml}\} = \widehat{p}_{ml}$ and $\mathrm{var}\{\mathsf{z}_{ml} \,|\, \mathsf{p}_{ml} = \widehat{p}_{ml}; \nu^p_{ml}\} = \nu^p_{ml}$, so that $\widehat{s}_{ml} = 0$ and $\nu^s_{ml} = 0$. This is expected, given that $\nu^s$ can be interpreted as an inverse residual variance and $\widehat{s}$ as a $\nu^s$-scaled residual. In summary, the PIAWGN model yields

$$\widehat{s}_{ml}(t) = \begin{cases} \frac{y_{ml} - \widehat{p}_{ml}(t)}{\nu^p_{ml}(t) + \nu^w} & (m,l) \notin \Omega \\ 0 & (m,l) \notin \Omega \end{cases} \tag{4.83}$$

$$\nu^s_{ml}(t) = \begin{cases} \frac{1}{\nu^p_{ml}(t) + \nu^w} & (m,l) \notin \Omega \\ 0 & (m,l) \notin \Omega \end{cases}. \tag{4.84}$$

When the PIAWGN model is combined with the scalar-variance approximations from Section 4.3.1, BiG-AMP simplifies considerably. To see this, we start by using $\nu^p(t)$ from (4.77) in place of $\nu^p_{ml}(t)$ in (4.83)-(4.84), resulting in

$$\widehat{\boldsymbol{S}}(t) = P_\Omega\left(\frac{\boldsymbol{Y} - \widehat{\boldsymbol{P}}(t)}{\nu^p(t) + \nu^w}\right) \tag{4.85}$$

$$\nu^s(t) = \frac{\delta}{\nu^w + \nu^p(t)}, \tag{4.86}$$

where $\delta \triangleq \frac{|\Omega|}{ML}$ denotes the fraction of observed entries and $P_\Omega : \mathbb{R}^{M\times L} \to \mathbb{R}^{M\times L}$ is the projection operator defined by

$$\big[P_\Omega(\boldsymbol{Z})\big]_{ml} \triangleq \begin{cases} z_{ml} & (m,l) \in \Omega \\ 0 & (m,l) \notin \Omega \end{cases}. \tag{4.87}$$

We can then write (R10) and (R12) as

$$\widehat{\boldsymbol{R}}(t) = \widehat{\boldsymbol{X}}(t)\left(1 - \frac{MN\nu^a(t)}{\|\widehat{\boldsymbol{A}}(t)\|_F^2}\right) + \frac{N}{\delta\|\widehat{\boldsymbol{A}}(t)\|_F^2}\widehat{\boldsymbol{A}}^{\mathsf{H}}(t)\widehat{\boldsymbol{V}}(t) \tag{4.88}$$

$$\widehat{\boldsymbol{Q}}(t) = \widehat{\boldsymbol{A}}(t)\left(1 - \frac{NL\nu^x(t)}{\|\widehat{\boldsymbol{X}}(t)\|_F^2}\right) + \frac{N}{\delta\|\widehat{\boldsymbol{X}}(t)\|_F^2}\widehat{\boldsymbol{V}}(t)\widehat{\boldsymbol{X}}^{\mathsf{H}}(t) \tag{4.89}$$

using (4.80)-(4.81) and (4.85)-(4.87) with

$$\widehat{\boldsymbol{V}}(t) \triangleq P_\Omega\big(\boldsymbol{Y} - \widehat{\boldsymbol{P}}(t)\big) \tag{4.90}$$

$$= P_\Omega\big(\boldsymbol{Y} - \overline{\boldsymbol{P}}(t)\big) + \overline{\nu}^p(t)\widehat{\boldsymbol{S}}(t-1) \tag{4.91}$$

$$= P_\Omega\big(\boldsymbol{Y} - \overline{\boldsymbol{P}}(t)\big) + \frac{\overline{\nu}^p(t)}{\nu^p(t-1)+\nu^w}\widehat{\boldsymbol{V}}(t-1), \tag{4.92}$$

since $P_\Omega$ is a projection operator, and using (R4) and (4.85).

Scalar-variance BiG-AMP under PIAWGN observations is summarized in Table 4.4. Note that the residual matrix $\widehat{\boldsymbol{U}}(t) \triangleq P_\Omega(\boldsymbol{Y} - \widehat{\boldsymbol{A}}(t)\widehat{\boldsymbol{X}}(t))$ needs to be computed and stored only at the observed entries $(m,l) \in \Omega$, leading to significant savings[15] when the observations are highly incomplete (i.e., $|\Omega| \ll ML$). The same is true for the Onsager-corrected residual, $\widehat{\boldsymbol{V}}(t)$. Thus, the algorithm in Table 4.4 involves only three (partial) matrix multiplies [in steps (R3p), (R8p), and (R10p), respectively], each of which can be computed using only $N|\Omega|$ scalar multiplies.

We note that Krzakala, Mézard, and Zdeborová recently proposed an AMP-based approach to blind calibration and dictionary learning [68] that bears close similarity[16]

[15]Similar computational savings also occur with incomplete non-Gaussian observations.

[16]The approach in [68] does not compute (or use) $\nu^p(t)$ as given in lines (R4p)-(R5p) of Table 4.4, but rather uses an empirical average of the squared Onsager-corrected residual in place of our $\nu^p(t)+\nu^w$ throughout their algorithm.

initialization:

$$\widehat{\boldsymbol{V}}(0) = \mathbf{0} \quad \text{(I1p)}$$

$$\text{choose} \quad \nu^x(1), \widehat{\boldsymbol{X}}(1), \nu^a(1), \widehat{\boldsymbol{A}}(1) \quad \text{(I2p)}$$

for $t = 1, \dots T_{\max}$

$$G_a(t) = \frac{N}{\delta \|\widehat{\boldsymbol{A}}(t)\|_F^2} \quad \text{(R1p)}$$

$$G_x(t) = \frac{N}{\delta \|\widehat{\boldsymbol{X}}(t)\|_F^2} \quad \text{(R2p)}$$

$$\widehat{\boldsymbol{U}}(t) = P_\Omega\big(\boldsymbol{Y} - \widehat{\boldsymbol{A}}(t)\widehat{\boldsymbol{X}}(t)\big) \quad \text{(R3p)}$$

$$\overline{\nu}^p(t) = \big(\tfrac{\nu^x(t)}{MG_a(t)} + \tfrac{\nu^a(t)}{LG_x(t)}\big)\tfrac{N}{\delta} \quad \text{(R4p)}$$

$$\nu^p(t) = \overline{\nu}^p(t) + N\nu^a(t)\nu^x(t) \quad \text{(R5p)}$$

$$\widehat{\boldsymbol{V}}(t) = \widehat{\boldsymbol{U}}(t) + \tfrac{\overline{\nu}^p(t)}{\nu^p(t-1)+\nu^w}\widehat{\boldsymbol{V}}(t-1) \quad \text{(R6p)}$$

$$\nu^r(t) = G_a(t)\big(\nu^p(t) + \nu^w\big) \quad \text{(R7p)}$$

$$\widehat{\boldsymbol{R}}(t) = (1 - M\delta\nu^a(t)G_a(t))\widehat{\boldsymbol{X}}(t) + G_a(t)\widehat{\boldsymbol{A}}^{\mathsf{H}}(t)\widehat{\boldsymbol{V}}(t) \quad \text{(R8p)}$$

$$\nu^q(t) = G_x(t)\big(\nu^p(t) + \nu^w\big) \quad \text{(R9p)}$$

$$\widehat{\boldsymbol{Q}}(t) = (1 - L\delta\nu^x(t)G_x(t))\widehat{\boldsymbol{A}}(t) + G_x(t)\widehat{\boldsymbol{V}}(t)\widehat{\boldsymbol{X}}^{\mathsf{H}}(t) \quad \text{(R10p)}$$

$$\nu^x(t+1) = \tfrac{1}{NL}\textstyle\sum_{n=1}^{N}\sum_{l=1}^{L} \operatorname{var}\{\mathsf{x}_{nl} \,|\, \boldsymbol{Y}; \widehat{r}_{nl}(t), \nu^r(t)\} \quad \text{(R11p)}$$

$$\forall n, l : \widehat{x}_{nl}(t+1) = \mathrm{E}\{\mathsf{x}_{nl} \,|\, \boldsymbol{Y}; \widehat{r}_{nl}(t), \nu^r(t)\} \quad \text{(R12p)}$$

$$\nu^a(t+1) = \tfrac{1}{MN}\textstyle\sum_{m=1}^{M}\sum_{n=1}^{N} \operatorname{var}\{\mathsf{a}_{mn}|\boldsymbol{Y}; \widehat{q}_{mn}(t), \nu^q(t)\} \quad \text{(R13p)}$$

$$\forall m, n : \widehat{a}_{mn}(t+1) = \mathrm{E}\{\mathsf{a}_{mn} \,|\, \boldsymbol{Y}; \widehat{q}_{mn}(t), \nu^q(t)\} \quad \text{(R14p)}$$

$$\text{if } \|\widehat{\boldsymbol{U}}(t) - \widehat{\boldsymbol{U}}(t-1)\|_F^2 \le \tau_{\text{BiG-AMP}}\|\widehat{\boldsymbol{U}}(t)\|_F^2, \text{ stop} \quad \text{(R15p)}$$

end

Table 4.4: Scalar-variance BiG-AMP with PIAWGN $p_{\mathsf{y}|\mathsf{z}}$

to BiG-AMP under the special case of AWGN-corrupted observations (i.e., $|\Omega| = ML$) and scalar variances. Their derivation differs significantly from that in Section 4.2 due to the many simplifications offered by this special case.

### 4.3.3 Zero-mean iid Gaussian Priors on A and X

In this section we will investigate the simplifications that result in the case that both $p_{\mathsf{a}_{mn}}$ and $p_{\mathsf{x}_{nl}}$ are zero-mean iid Gaussian, i.e.,

$$p_{\mathsf{x}_{nl}}(x) = \mathcal{N}(x; 0, \nu_0^x) \ \forall n, l \tag{4.93}$$

$$p_{\mathsf{a}_{mn}}(a) = \mathcal{N}(a; 0, \nu_0^a) \ \forall m, n, \tag{4.94}$$

which, as will be discussed later, is appropriate for matrix completion. In this case, straightforward calculations reveal that $\mathrm{E}\{\mathsf{x}_{nl} \,|\, \mathsf{r}_{nl} = \widehat{r}_{nl}; \nu_{nl}^r\} = \widehat{r}_{nl}\nu_0^x/(\nu_{nl}^r + \nu_0^x)$

initialization:

$$\widehat{\boldsymbol{V}}(0) = \mathbf{0} \quad \text{(I1i)}$$

$$\text{choose} \quad \nu^x(1), \widehat{\boldsymbol{X}}(1), \nu^a(1), \widehat{\boldsymbol{A}}(1) \quad \text{(I2i)}$$

for $t = 1, \dots T_{\max}$

$$G_a(t) = \frac{N}{\delta \|\widehat{\boldsymbol{A}}(t)\|_F^2} \quad \text{(R1i)}$$

$$G_x(t) = \frac{N}{\delta \|\widehat{\boldsymbol{X}}(t)\|_F^2} \quad \text{(R2i)}$$

$$\widehat{\boldsymbol{U}}(t) = P_\Omega\big(\boldsymbol{Y} - \widehat{\boldsymbol{A}}(t)\widehat{\boldsymbol{X}}(t)\big) \quad \text{(R3i)}$$

$$\overline{\nu}^p(t) = \big(\tfrac{\nu^x(t)}{MG_a(t)} + \tfrac{\nu^a(t)}{LG_x(t)}\big)\tfrac{N}{\delta} \quad \text{(R4i)}$$

$$\nu^p(t) = \overline{\nu}^p(t) + N\nu^a(t)\nu^x(t) \quad \text{(R5i)}$$

$$\widehat{\boldsymbol{V}}(t) = \widehat{\boldsymbol{U}}(t) + \tfrac{\overline{\nu}^p(t)}{\nu^p(t-1)+\nu^w}\widehat{\boldsymbol{V}}(t-1) \quad \text{(R6i)}$$

$$\nu^r(t) = G_a(t)\big(\nu^p(t) + \nu^w\big) \quad \text{(R7i)}$$

$$\nu^q(t) = G_x(t)\big(\nu^p(t) + \nu^w\big) \quad \text{(R8i)}$$

$$\nu^x(t+1) = \big(\tfrac{1}{\nu^r(t)} + \tfrac{1}{\nu_0^x}\big)^{-1} \quad \text{(R9i)}$$

$$\widehat{\boldsymbol{X}}(t+1) = \tfrac{\nu^x(t+1)}{\nu^r(t)}\big((1 - M\delta\nu^a(t)G_a(t))\widehat{\boldsymbol{X}}(t) + G_a(t)\widehat{\boldsymbol{A}}^{\mathsf{H}}(t)\widehat{\boldsymbol{V}}(t)\big) \quad \text{(R10i)}$$

$$\nu^a(t+1) = \big(\tfrac{1}{\nu^q(t)} + \tfrac{1}{\nu_0^a}\big)^{-1} \quad \text{(R11i)}$$

$$\widehat{\boldsymbol{A}}(t+1) = \tfrac{\nu^a(t+1)}{\nu^q(t)}\big((1 - L\delta\nu^x(t)G_x(t))\widehat{\boldsymbol{A}}(t) + G_x(t)\widehat{\boldsymbol{V}}(t)\widehat{\boldsymbol{X}}^{\mathsf{H}}(t)\big) \quad \text{(R12i)}$$

$$\text{if } \|\widehat{\boldsymbol{U}}(t) - \widehat{\boldsymbol{U}}(t-1)\|_F^2 \le \tau_{\text{BiG-AMP}}\|\widehat{\boldsymbol{U}}(t)\|_F^2, \text{ stop} \quad \text{(R13i)}$$

end

Table 4.5: BiG-AMP-Lite: Scalar-variance, PIAWGN, Gaussian $p_{\mathsf{x}}$ and $p_{\mathsf{a}}$

and $\text{var}\{\mathsf{x}_{nl} \,|\, \mathsf{r}_{nl} = \widehat{r}_{nl}; \nu^r_{nl}\}) = \nu^x_0\nu^r_{nl}/(\nu^r_{nl} + \nu^x_0)$ and, similarly, that $\text{E}\{\mathsf{a}_{mn} \,|\, \mathsf{q}_{mn} = \widehat{q}_{mn}; \nu^q_{mn}\} = \widehat{q}_{mn}\nu^a_0/(\nu^q_{mn} + \nu^a_0)$ and $\text{var}\{\mathsf{a}_{mn} \,|\, \mathsf{q}_{mn} = \widehat{q}_{mn}, \nu^q_{mn}\} = \nu^a_0\nu^q_{mn}/(\nu^q_{mn} + \nu^a_0)$. Combining these iid Gaussian simplifications with the scalar-variance simplifications from Section 4.3.1 yields an algorithm whose computational cost is dominated by three matrix multiplies per iteration, each with a cost of $MNL$ scalar multiplies. The precise number of multiplies it consumes depends on the assumed likelihood model that determines steps (R7g)-(R8g).

Additionally incorporating the PIAWGN observations from Section 4.3.2 reduces the cost of the three matrix multiplies to only $N|\Omega|$ scalar multiplies each, and yields the "BiG-AMP-Lite" algorithm summarized in Table 4.5, consuming $(3N + 5)|\Omega| + 3(MN + NL) + 29$ multiplies per iteration.

## 4.4 Adaptive Damping

The approximations made in the BiG-AMP derivation presented in Section 4.2 were well-justified in the large system limit, i.e., the case where $M, N, L \to \infty$ with fixed $\frac{M}{N}$ and $\frac{L}{N}$. In practical applications, however, these dimensions (especially $N$) are finite, and hence the algorithm presented in Section 4.2 may diverge. In case of compressive sensing, the use of "damping" with GAMP yields provable convergence guarantees with arbitrary matrices [18]. Here, we propose to incorporate damping into BiG-AMP. Moreover, we propose to *adapt* the damping of these variables to ensure that a particular cost criterion decreases monotonically (or near-monotonically), as described in the sequel. The specific damping strategy that we adopt is similar to that described in [74] and coded in [19].

### 4.4.1 Damping

In BiG-AMP, the iteration-$t$ damping factor $\beta(t) \in (0, 1]$ is used to slow the evolution of certain variables, namely $\overline{\nu}^p_{ml}$, $\nu^p_{ml}$, $\nu^s_{ml}$, $\widehat{s}_{ml}$, $\widehat{x}_{nl}$, and $\widehat{a}_{mn}$. To do this,

steps (R1), (R3), (R7), and (R8) in Table 4.3 are replaced with

$$\overline{\nu}^p_{ml}(t) = \beta(t)\bigg(\sum_{n=1}^{N} |\widehat{a}_{mn}(t)|^2\nu^x_{nl}(t) + \nu^a_{mn}(t)|\widehat{x}_{nl}(t)|^2\bigg) + (1-\beta(t))\overline{\nu}^p_{ml}(t-1) \tag{4.95}$$

$$\nu^p_{ml}(t) = \beta(t)\bigg(\overline{\nu}^p_{ml}(t) + \sum_{n=1}^{N} \nu^a_{mn}(t)\nu^x_{nl}(t)\bigg) + (1-\beta(t))\nu^p_{ml}(t-1) \tag{4.96}$$

$$\nu^s_{ml}(t) = \beta(t)\big((1-\nu^z_{ml}(t)/\nu^p_{ml}(t))/\nu^p_{ml}(t)\big) + (1-\beta(t))\nu^s_{ml}(t-1) \tag{4.97}$$

$$\widehat{s}_{ml}(t) = \beta(t)\big(\widehat{z}_{ml}(t) - \widehat{p}_{ml}(t))/\nu^p_{ml}(t)\big) + (1-\beta(t))\widehat{s}_{ml}(t-1), \tag{4.98}$$

and the following are inserted between (R8) and (R9):

$$\overline{x}_{nl}(t) = \beta(t)\widehat{x}_{nl}(t) + (1-\beta(t))\overline{x}_{nl}(t-1) \tag{4.99}$$

$$\overline{a}_{mn}(t) = \beta(t)\widehat{a}_{mn}(t) + (1-\beta(t))\overline{a}_{mn}(t-1). \tag{4.100}$$

The newly defined state variables $\overline{x}_{nl}(t)$ and $\overline{a}_{mn}(t)$ are then used in place of $\widehat{x}_{nl}(t)$ and $\widehat{a}_{mn}(t)$ in steps (R9)-(R12) [but not (R1)-(R2)] of Table 4.3. A similar approach can be used for the algorithm in Table 4.4 (with the damping applied to $\widehat{\boldsymbol{V}}(t)$ instead of $\widehat{\boldsymbol{S}}(t)$) and those in Table 4.5. Notice that, when $\beta(t)=1$, the damping has no effect, whereas when $\beta(t)=0$, all quantities become frozen in $t$.

### 4.4.2 Adaptive Damping

The idea behind adaptive damping is to monitor a chosen cost criterion $J(t)$ and decrease $\beta(t)$ when the cost has not decreased sufficiently[17] relative to $\{J(\tau)\}_{\tau=t-1-T}^{t-1}$

[17]The following adaptation procedure is borrowed from GAMPmatlab [19], where it has been established to work well in the context of GAMP-based compressive sensing. When the current

for some "step window" $T \geq 0$. This mechanism allows the cost criterion to increase over short intervals of $T$ iterations and in this sense is similar to the procedure used by SpaRSA [75]. We now describe how the cost criterion $J(t)$ is constructed, building on ideas in [20] that were reviewed in Section 2.8.

Notice that, for fixed observations $\boldsymbol{Y}$, the joint posterior pdf solves the (trivial) KL-divergence minimization problem

$$p_{\mathbf{X},\mathbf{A}|\mathbf{Y}} = \arg\min_{b_{\mathbf{X},\mathbf{A}}} D(b_{\mathbf{X},\mathbf{A}} \| p_{\mathbf{X},\mathbf{A}|\mathbf{Y}}). \tag{4.101}$$

The factorized form (4.5) of the posterior allows us to write

$$\begin{aligned} & D(b_{\mathbf{X},\mathbf{A}} \| p_{\mathbf{X},\mathbf{A}|\mathbf{Y}}) - \log p_{\mathbf{Y}}(\boldsymbol{Y}) \\ &= \int_{\boldsymbol{A},\boldsymbol{X}} b_{\mathbf{X},\mathbf{A}}(\boldsymbol{A},\boldsymbol{X}) \log \frac{b_{\mathbf{X},\mathbf{A}}(\boldsymbol{A},\boldsymbol{X})}{p_{\mathbf{Y}|\mathbf{Z}}(\boldsymbol{Y} \mid \boldsymbol{A}\boldsymbol{X})\, p_{\mathbf{X}}(\boldsymbol{X})\, p_{\mathbf{A}}(\boldsymbol{A})} && (4.102) \\ &= D(b_{\mathbf{X},\mathbf{A}} \| p_{\mathbf{A}} p_{\mathbf{X}}) - \int_{\boldsymbol{A},\boldsymbol{X}} b_{\mathbf{X},\mathbf{A}}(\boldsymbol{A},\boldsymbol{X}) \log p_{\mathbf{Y}|\mathbf{Z}}(\boldsymbol{Y} \mid \boldsymbol{A}\boldsymbol{X}) && (4.103) \end{aligned}$$

Equations (4.101) and (4.103) then imply that

$$\begin{aligned} p_{\mathbf{X},\mathbf{A}|\mathbf{Y}} &= \arg\min_{b_{\mathbf{X},\mathbf{A}}} J(b_{\mathbf{X},\mathbf{A}}) && (4.104) \\ J(b_{\mathbf{X},\mathbf{A}}) &\triangleq D(b_{\mathbf{X},\mathbf{A}} \| p_{\mathbf{A}} p_{\mathbf{X}}) - \mathrm{E}_{b_{\mathbf{X},\mathbf{A}}} \big\{ \log p_{\mathbf{Y}|\mathbf{Z}}(\boldsymbol{Y} \mid \mathbf{A}\mathbf{X}) \big\}. && (4.105) \end{aligned}$$

To judge whether a given time-$t$ BiG-AMP approximation "$b_{\mathbf{X},\mathbf{A}}(t)$" of the joint posterior $p_{\mathbf{X},\mathbf{A}|\mathbf{Y}}$ is better than the previous approximation $b_{\mathbf{X},\mathbf{A}}(t-1)$, one could in principle plug the posterior approximation expressions (4.71)-(4.72) into (4.105) and then check whether $J(b_{\mathbf{X},\mathbf{A}}(t)) < J(b_{\mathbf{X},\mathbf{A}}(t-1))$. But, since the expectation in (4.105)

cost $J(t)$ is not smaller than the largest cost in the most recent `stepWindow` iterations, then the "step" is deemed unsuccessful, the damping factor $\beta(t)$ is reduced by the factor `stepDec`, and the step is attempted again. These attempts continue until either the cost criterion decreases or the damping factor reaches `stepMin`, at which point the step is considered successful, or the iteration count exceeds $T_{\max}$ or the damping factor reaches `stepTol`, at which point the algorithm terminates. When a step is deemed successful, the damping factor is increased by the factor `stepInc`, up to the allowed maximum value `stepMax`.

is difficult to evaluate, we approximate the cost (4.105) by using, in place of $\mathbf{AX}$, an independent Gaussian matrix[18] whose component means and variances are matched to those of $\mathbf{AX}$. Taking the joint BiG-AMP posterior approximation $b_{\mathbf{X},\mathbf{A}}(t)$ to be the product of the marginals from (4.71)-(4.72), the resulting component means and variances are

$$\begin{aligned}
\mathrm{E}_{b_{\mathbf{X},\mathbf{A}}(t)}\{[\mathbf{AX}]_{ml}\} &= \sum_n \mathrm{E}_{b_{\mathbf{X},\mathbf{A}}(t)}\{\mathsf{a}_{mn}\}\,\mathrm{E}_{b_{\mathbf{X},\mathbf{A}}(t)}\{\mathsf{x}_{nl}\} && (4.106)\\
&= \sum_n \widehat{a}_{mn}(t)\widehat{x}_{nl}(t) \;= \overline{p}_{ml}(t) && (4.107)\\
\mathrm{var}_{b_{\mathbf{X},\mathbf{A}}(t)}\{[\mathbf{AX}]_{ml}\} &= \sum_n \widehat{a}^2_{mn}(t)\nu^x_{nl}(t) + \nu^a_{mn}(t)\widehat{x}^2_{nl}(t)\\
&\quad + \nu^a_{mn}(t)\nu^x_{nl}(t) && (4.108)\\
&= \nu^p_{ml}(t). && (4.109)
\end{aligned}$$

In this way, the approximate iteration-$t$ cost becomes

$$\begin{aligned}
\widehat{J}(t) = &\sum_{n,l} D\Big(p_{\mathsf{x}_{nl}|\mathsf{r}_{nl}}\big(\,\cdot\,\big|\,\widehat{r}_{nl}(t);\nu^r_{nl}(t)\big)\Big\|\,p_{\mathsf{x}_{nl}}(\cdot)\Big) \qquad (4.110)\\
&+ \sum_{m,n} D\Big(p_{\mathsf{a}_{mn}|\mathsf{q}_{mn}}\big(\,\cdot\,\big|\,\widehat{q}_{mn}(t);\nu^q_{mn}(t)\big)\Big\|\,p_{\mathsf{a}_{mn}}(\cdot)\Big)\\
&- \sum_{m,l} \mathrm{E}_{\mathsf{z}_{ml}\sim\mathcal{N}(\overline{p}_{ml}(t);\nu^p_{ml}(t))}\big\{\log p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml}\,|\,\mathsf{z}_{ml})\big\}.
\end{aligned}$$

Intuitively, the first term in (4.110) penalizes the deviation between the (BiG-AMP approximated) posterior and the assumed prior on $\mathbf{X}$, the second penalizes the deviation between the (BiG-AMP approximated) posterior and the assumed prior on $\mathbf{A}$, and the third term rewards highly likely estimates $\mathbf{Z}$.

[18]The GAMP work [20] uses a similar approximation.

## 4.5 Parameter Tuning and Rank Selection

### 4.5.1 Parameter Tuning via Expectation Maximization

Recall that BiG-AMP requires the specification of priors $p_{\mathbf{X}}(\boldsymbol{X}) = \prod_{n,l} p_{\mathsf{x}_{nl}}(x_{nl})$, $p_{\mathbf{A}}(\boldsymbol{A}) = \prod_{m,n} p_{\mathsf{a}_{mn}}(a_{mn})$, and $p_{\mathbf{Y}|\mathbf{Z}}(\boldsymbol{Y}|\boldsymbol{Z}) = \prod_{m,l} p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml}|z_{ml})$. In practice, although one may know appropriate families for these distributions, the exact parameters that govern them are generally unknown. For example, one may have good reason to believe apriori that the observations are AWGN corrupted, justifying the choice $p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml}|z_{ml}) = \mathcal{N}(y_{ml}; z_{ml}, \nu^w)$, but the noise variance $\nu^w$ may be unknown. In this section, we outline a methodology that takes a given set of BiG-AMP parameterized priors $\{p_{\mathsf{x}_{nl}}(\cdot;\boldsymbol{\theta}), p_{\mathsf{a}_{mn}}(\cdot;\boldsymbol{\theta}), p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml}|\cdot;\boldsymbol{\theta})\}_{\forall m,n,l}$ and tunes the parameter vector $\boldsymbol{\theta}$ using an expectation-maximization (EM) [61] based approach, with the goal of maximizing the likelihood, i.e., finding $\widehat{\boldsymbol{\theta}} \triangleq \arg\max_{\boldsymbol{\theta}} p_{\mathbf{Y}}(\boldsymbol{Y};\boldsymbol{\theta})$. The approach presented here can be considered as a generalization of the GAMP-based work [62] to BiG-AMP.

Taking $\mathbf{X}$, $\mathbf{A}$, and $\mathbf{Z}$ to be the hidden variables, the EM recursion can be written as [61]

$$\begin{aligned}
\widehat{\boldsymbol{\theta}}^{k+1} &= \arg\max_{\boldsymbol{\theta}} \mathrm{E}\left\{ \log p_{\mathbf{X},\mathbf{A},\mathbf{Z},\mathbf{Y}}(\mathbf{X},\mathbf{A},\mathbf{Z},\mathbf{Y};\boldsymbol{\theta}) \,\middle|\, \boldsymbol{Y}; \widehat{\boldsymbol{\theta}}^k \right\} \\
&= \arg\max_{\boldsymbol{\theta}} \Bigg\{ \sum_{n,l} \mathrm{E}\left\{ \log p_{\mathsf{x}_{nl}}(\mathsf{x}_{nl};\boldsymbol{\theta}) \,\middle|\, \boldsymbol{Y}; \widehat{\boldsymbol{\theta}}^k \right\} \qquad (4.111) \\
&\quad + \sum_{m,n} \mathrm{E}\left\{ \log p_{\mathsf{a}_{mn}}(\mathsf{a}_{mn};\boldsymbol{\theta}) \,\middle|\, \boldsymbol{Y}; \widehat{\boldsymbol{\theta}}^k \right\} \\
&\quad + \sum_{m,l} \mathrm{E}\left\{ \log p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml} \,|\, \mathsf{z}_{ml};\boldsymbol{\theta}) \,\middle|\, \boldsymbol{Y}; \widehat{\boldsymbol{\theta}}^k \right\} \Bigg\}
\end{aligned}$$

where for (4.111) we used the fact $p_{\mathbf{X},\mathbf{A},\mathbf{Z},\mathbf{Y}}(\mathbf{X},\mathbf{A},\mathbf{Z},\mathbf{Y};\boldsymbol{\theta}) = p_{\mathbf{X}}(\mathbf{X};\boldsymbol{\theta}) p_{\mathbf{A}}(\mathbf{A};\boldsymbol{\theta}) p_{\mathbf{Y}|\mathbf{Z}}(\mathbf{Y}|\mathbf{Z};\boldsymbol{\theta})\, 1_{\mathbf{Z}=\mathbf{A}\mathbf{X}}$ and the factorizability of $p_{\mathbf{X}}$, $p_{\mathbf{A}}$, and $p_{\mathbf{Y}|\mathbf{Z}}$. As can

be seen from (4.111), knowledge of the marginal posteriors $\{p_{\mathsf{x}_{nl}|\mathbf{Y}}, p_{\mathsf{a}_{mn}|\mathbf{Y}}, p_{\mathsf{z}_{ml}|\mathbf{Y}}\}_{\forall m,n,l}$ is sufficient to compute the EM update. Since the exact marginal posteriors are unknown, we employ BiG-AMP's approximations from (4.71), (4.72), and (4.35) for approximate EM. In addition, we adopt the "incremental" update strategy from [76], where the maximization over $\boldsymbol{\theta}$ is performed one element at a time while holding the others fixed.

As a concrete example, consider updating the noise variance $\nu^w$ under the PI-AWGN model (4.82). Equation (4.111) suggests

$$(\nu^w)^{k+1} = \arg\max_{\nu^w} \sum_{(m,l)\in\Omega} \int_{z_{ml}} p_{\mathsf{z}_{ml}|\mathbf{Y}}(z_{ml}|\boldsymbol{Y}) \times \log \mathcal{N}(y_{ml}; z_{ml}, \nu^w), \tag{4.112}$$

where the true marginal posterior $p_{\mathsf{z}_{ml}|\mathbf{Y}}(\cdot|\boldsymbol{Y})$ is replaced with the most recent BiG-AMP approximation $p_{\mathsf{z}_{ml}|\mathsf{p}_{ml}}(\cdot|\widehat{p}_{ml}(T_{\max}); \nu^p_{ml}(T_{\max}), \widehat{\boldsymbol{\theta}}^k)$, where "most recent" is with respect to both EM and BiG-AMP iterations. Zeroing the derivative of the sum in (4.112) with respect to $\nu^w$,

$$(\nu^w)^{k+1} = \frac{1}{|\Omega|} \sum_{(m,l)\in\Omega} \big(y_{ml} - \widehat{z}_{ml}(T_{\max})\big)^2 + \nu^z_{ml}(T_{\max}), \tag{4.113}$$

where $\widehat{z}_{ml}(t)$ and $\nu^z_{ml}(t)$ are the BiG-AMP approximated posterior mean and variance from (4.33)-(4.34).

The overall procedure can be summarized as follows. From a suitable initialization $\widehat{\boldsymbol{\theta}}^0$, BiG-AMP is run using the priors $\{p_{\mathsf{x}_{nl}}(\cdot;\widehat{\boldsymbol{\theta}}^0), p_{\mathsf{a}_{mn}}(\cdot;\widehat{\boldsymbol{\theta}}^0), p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml}|\cdot;\widehat{\boldsymbol{\theta}}^0)\}_{\forall m,n,l}$ and iterated to completion, yielding approximate marginal posteriors on $\{\mathsf{x}_{nl}, \mathsf{a}_{mn}, \mathsf{z}_{ml}\}_{\forall m,n,l}$. These posteriors are used in (4.111) to update the parameters $\boldsymbol{\theta}$ one element at a time, yielding $\widehat{\boldsymbol{\theta}}^1$. BiG-AMP is then run using the priors $\{p_{\mathsf{x}_{nl}}(\cdot;\widehat{\boldsymbol{\theta}}^1), p_{\mathsf{a}_{mn}}(\cdot;\widehat{\boldsymbol{\theta}}^1), p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml}|\cdot;\widehat{\boldsymbol{\theta}}^1)\}_{\forall m,n,l}$, and so on. A detailed discussion in

the context of GAMP, along with explicit update equations for the parameters of Bernoulli-Gaussian-mixture pdfs, can be found in [62].

### 4.5.2 Rank Selection

BiG-AMP and EM-BiG-AMP, as described up to this point, require the specification of the rank $N$, i.e., the number of columns in $\mathbf{A}$ (and rows in $\mathbf{X}$) in the matrix factorization $\mathbf{Z} = \mathbf{AX}$. Since, in many applications, the best choice of $N$ is difficult to specify in advance, we now describe two procedures to estimate $N$ from the data $\boldsymbol{Y}$, building on well-known rank-selection procedures.

**Penalized log-likelihood maximization**

Consider a set of possible models $\{\mathcal{H}_N\}_{N=1}^{\overline{N}}$ for the observation $\boldsymbol{Y}$ where, under $\mathcal{H}_N$, EM-BiG-AMP estimates $\boldsymbol{\Theta}_N = \{\mathbf{A}_N, \mathbf{X}_N, \boldsymbol{\theta}\}$. Here, the subscripts on $\mathbf{A}_N$ and $\mathbf{X}_N$ indicate the restriction to $N$ columns and rows, $\boldsymbol{\theta}$ refers to the vector of parameters defined in Section 4.5.1, and the subscript on $\boldsymbol{\Theta}_N$ indicates the dependence of the overall number of parameters in $\boldsymbol{\Theta}_N$ with the rank $N$. Because the selection rule $\widehat{N} = \arg\max_N p_{\mathbf{Y}}(\boldsymbol{Y}; \mathcal{H}_N)$ is typically intractable, several well-known rules of the form

$$\widehat{N} = \underset{N=1,\ldots,\overline{N}}{\arg\max}\; 2\log p_{\mathbf{Y}|\boldsymbol{\Theta}_N}(\boldsymbol{Y} \mid \widehat{\boldsymbol{\Theta}}_N) - \eta(N) \tag{4.114}$$

have been developed, such as the Bayesian Information Criterion (BIC) and Akaike's Information Criterion (AIC) [77]. In (4.114), $\widehat{\boldsymbol{\Theta}}_N$ is the ML estimate of $\boldsymbol{\Theta}_N$ under $\boldsymbol{Y}$, and $\eta(\cdot)$ is a penalty function that depends on the *effective* number of scalar parameters $N_{\text{eff}}$ estimated under model $\mathcal{H}_N$ (which depends on $N$) and possibly on the number of scalar parameters $|\Omega|$ that make up the observation $\boldsymbol{Y}$.

Applying this methodology to EM-BiG-AMP, where $p_{\mathbf{Y}|\mathbf{\Theta}_N}(\boldsymbol{Y} \mid \mathbf{\Theta}_N) = p_{\mathbf{Y}|\mathbf{Z}}(\boldsymbol{Y} \mid \boldsymbol{A}_N \boldsymbol{X}_N; \boldsymbol{\theta})$, we obtain the rank-selection rule

$$\widehat{N} = \underset{N=1,\ldots,\overline{N}}{\arg\max}\; 2 \log p_{\mathbf{Y}|\mathbf{Z}}(\boldsymbol{Y} \mid \widehat{\boldsymbol{A}}_N \widehat{\boldsymbol{X}}_N; \widehat{\boldsymbol{\theta}}) - \eta(N). \tag{4.115}$$

Since $N_{\text{eff}}$ depends on the application (e.g., matrix completion, robust PCA, dictionary learning), detailed descriptions of $\eta(\cdot)$ are deferred to the discussions of specific examples, such as in Section 4.6.5.

To perform the maximization over $N$ in (4.115), we start with a small hypothesis $N_1$ and run EM-BiG-AMP to completion, generating the (approximate) MMSE estimates $\widehat{\boldsymbol{A}}_{N_1}, \widehat{\boldsymbol{X}}_{N_1}$ and ML estimate $\widehat{\boldsymbol{\theta}}$, which are then used to evaluate[19] the penalized log-likelihood in (4.115). The $N$ hypothesis is then increased by a fixed value (i.e., $N_2 = N_1 + \texttt{rankStep}$), initializations of $(\boldsymbol{A}_{N_2}, \boldsymbol{X}_{N_2}, \boldsymbol{\theta})$ are chosen based on the previously computed $(\widehat{\boldsymbol{A}}_{N_1}, \widehat{\boldsymbol{X}}_{N_1}, \widehat{\boldsymbol{\theta}})$, and EM-BiG-AMP is run to completion, yielding estimates $(\widehat{\boldsymbol{A}}_{N_2}, \widehat{\boldsymbol{X}}_{N_2}, \widehat{\boldsymbol{\theta}})$ with which the penalized likelihood is again evaluated. This process continues until either the value of the penalized log-likelihood decreases, in which case $\widehat{N}$ is set at the previous (i.e., maximizing) hypothesis of $N$, or the maximum-allowed rank $\overline{N}$ is reached.

**Rank contraction**

We now describe an alternative rank-selection procedure that is appropriate when $\mathbf{Z}$ has a "cliff" in its singular value profile and which is reminiscent of that used in LMaFit [42]. In this approach, EM-BiG-AMP is initially configured to use the maximum-allowed rank, i.e., $N = \overline{N}$. After the first EM iteration, the singular values $\{\sigma_n\}$ of the estimate $\widehat{\boldsymbol{X}}$ and the corresponding pairwise ratios $R_n = \sigma_n / \sigma_{n+1}$

[19] Since we compute approximate MMSE estimates rather than ML estimates, we are in fact evaluating a lower bound on the penalized log-likelihood.

are computed,[20] from which a candidate rank estimate $\widehat{N} = \arg\max_n R_n$ is identified, corresponding to the largest gap in successive singular values. However, this candidate is accepted only if this maximizing ratio exceeds the average ratio by the user-specified parameter $\tau_{\text{MOS}}$ (e.g., $\tau_{\text{MOS}} = 5$), i.e., if

$$R_{\widehat{N}} > \frac{\tau_{\text{MOS}}}{\overline{N} - 2} \sum_{i \neq \widehat{N}} R_i, \tag{4.116}$$

and if $\widehat{N}/\overline{N}$ is sufficiently small. Increasing $\tau_{\text{MOS}}$ makes the approach less prone to selecting an erroneous rank during the first few iterations, but making the value too large prevents the algorithm from detecting small gaps between the singular values. If $\widehat{N}$ is accepted, then the matrices $\mathbf{A}$ and $\mathbf{X}$ are pruned to size $\widehat{N}$ and EM-BiG-AMP is run to convergence. If not, EM-BiG-AMP is run for one more iteration, after which a new candidate $\widehat{N}$ is identified and checked for acceptance, and so on.

In many cases, a rank candidate is accepted after a small number of iterations, and thus only a few SVDs need be computed. This procedure has the advantage of running EM-BiG-AMP to convergence only once, rather than several times under different hypothesized ranks. However, when the singular values of $\mathbf{Z}$ decay smoothly, this procedure can mis-estimate the rank, as discussed in [42].

## 4.6 Matrix Completion

In this and the next two sections, we detail the application of BiG-AMP to the problems of matrix completion (MC), robust principle components analysis (RPCA), and dictionary learning (DL), respectively. For each application, we discuss the BiG-AMP's choice of matrix representation, priors, likelihood, initialization, adaptive damping, EM-driven parameter learning, and rank-selection. Also, for each

[20] In some cases the singular values of $\widehat{\boldsymbol{A}}$ could be used instead.

application, we provide an extensive empirical study comparing BiG-AMP to state-of-the-art solvers on both synthetic and real-world datasets. These results demonstrate that BiG-AMP yields excellent reconstruction performance (often best in class) while maintaining competitive runtimes. For each application of BiG-AMP discussed in the sequel, we recommend numerical settings for necessary parameter values, as well as initialization strategies when appropriate. Although we cannot guarantee that our recommendations are universally optimal, they worked well for the range of problems considered in this chapter, and we conjecture that they offer a useful starting point for further experimentation. Nevertheless, modifications may be appropriate when applying BiG-AMP outside the range of problems considered here. Our BiG-AMP Matlab code can be found as part of the GAMPmatlab package at `https://sourceforge.net/projects/gampmatlab/`, including examples of BiG-AMP applied to the MC, RPCA, and DL problems.

### 4.6.1 Problem setup

In matrix completion (MC) [78], one seeks to recover a rank-$N \ll \min(M, L)$ matrix $\mathbf{Z} \in \mathbb{R}^{M \times L}$ after observing a fraction $\delta = \frac{|\Omega|}{ML}$ of its (possibly noise-corrupted) entries, where $\Omega$ denotes the set of observations.

BiG-AMP approaches the MC problem by modeling the complete matrix $\mathbf{Z}$ as the product $\mathbf{Z} = \mathbf{AX}$ of random matrices $\mathbf{A} \in \mathbb{R}^{M \times N}$ and $\mathbf{X} \in \mathbb{R}^{N \times L}$ with priors of the decoupled form in (4.1)-(4.2), where $\mathbf{Z}$ is probabilistically related to the observed matrix $\boldsymbol{Y}$ through a likelihood $p_{\mathbf{Y}|\mathbf{Z}}(\boldsymbol{Y} \,|\, \mathbf{Z})$ of the decoupled form in (4.3). To finally

perform MC, BiG-AMP infers $\mathbf{A}$ and $\mathbf{X}$ from $\mathbf{Y}$ under the above model. The corresponding estimates $\widehat{\boldsymbol{A}}$ and $\widehat{\boldsymbol{X}}$ can then be multiplied to yield an estimate $\widehat{\boldsymbol{Z}} = \widehat{\boldsymbol{A}}\widehat{\boldsymbol{X}}$ of the noiseless complete matrix $\mathbf{Z}$.

As in several existing Bayesian approaches to matrix completion (e.g., [52,79–81]), we choose Gaussian priors for the factors $\mathbf{A}$ and $\mathbf{X}$. Although EM-BiG-AMP readily supports the use of priors with row- and/or column-dependent parameters, we focus on simple iid priors of the form

$$p_{\mathsf{a}_{mn}}(a) = \mathcal{N}(a; 0, 1) \ \forall m, n \tag{4.117}$$

$$p_{\mathsf{x}_{nl}}(x) = \mathcal{N}(x; \widehat{x}_0, \nu_0^x) \ \forall n, l, \tag{4.118}$$

where the mean and variance in (4.118) can be tuned using EM-BiG-AMP, as described in the sequel, and where the variance in (4.117) is fixed to avoid a scaling ambiguity between $\mathbf{A}$ and $\mathbf{X}$. Section 4.6.6 demonstrates that this simple approach is effective in attacking several MC problems of interest. Assuming the observation noise to be additive and Gaussian, we then choose the PIAWGN model from (4.82) for the likelihood $p_{\mathbf{Y}|\mathbf{Z}}$ given by

$$p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml} \,|\, z_{ml}) = \begin{cases} \mathcal{N}(y_{ml}; z_{ml}, \nu^w) & (m,l) \in \Omega \\ 1_{y_{ml}} & (m,l) \notin \Omega. \end{cases} \tag{4.119}$$

Note that, by using (4.117)-(4.118) with $\widehat{x}_0 = 0$ and the scalar-variance approximation from Section 4.3.1, the BiG-AMP algorithm from Table 4.3 reduces to the simpler BiG-AMP-Lite algorithm from Table 4.5 with $\nu_0^a = 1$.

### 4.6.2 Initialization

In most cases we advocate initializing the BiG-AMP quantities $\widehat{\boldsymbol{X}}(1)$ and $\widehat{\boldsymbol{A}}(1)$ using random draws from the priors $p_{\mathbf{X}}$ and $p_{\mathbf{A}}$, although setting either $\widehat{\boldsymbol{X}}(1)$ or $\widehat{\boldsymbol{A}}(1)$

at zero also seems to perform well in the MC application. Although it is also possible to use SVD-based initializations of $\widehat{\boldsymbol{X}}(1)$ and $\widehat{\boldsymbol{A}}(1)$ (i.e., for SVD $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{D}^T$, set $\widehat{\boldsymbol{A}}(1) = \boldsymbol{U}\boldsymbol{\Sigma}^{1/2}$ and $\widehat{\boldsymbol{X}}(1) = \boldsymbol{\Sigma}^{1/2}\boldsymbol{D}^T$) as done in LMaFit [42] and VSBL [53], experiments suggest that the extra computation required is rarely worthwhile for BiG-AMP.

As for the initializations $\nu^x_{nl}(1)$ and $\nu^a_{mn}(1)$, we advocate setting them at 10 times the prior variances in (4.117)-(4.118), which has the effect of weighting the measurements $\boldsymbol{Y}$ more than the priors $p_{\mathsf{X}}, p_{\mathsf{A}}$ during the first few iterations.

### 4.6.3 Adaptive damping

For the assumed likelihood (4.119) and priors (4.117)-(4.118), the adaptive-damping cost criterion $\widehat{J}(t)$ described in Section 4.4.2 reduces to

$$\begin{aligned}\widehat{J}(t) = &\frac{1}{2}\sum_{n,l}\left[\log\frac{\nu^x_0}{\nu^x(t)} + \left(\frac{\nu^x(t)}{\nu^x_0} - 1\right) + \frac{(\widehat{x}_{nl}(t) - \widehat{x}_0)^2}{\nu^x_0}\right] \\ &+ \frac{1}{2}\sum_{m,n}\left[\log\frac{1}{\nu^a(t)} + \left(\nu^a(t) - 1\right) + \widehat{a}^2_{mn}(t)\right] \\ &+ \frac{1}{\nu^w}\left(\frac{1}{2}\sum_{(m,l)\in\Omega}(y_{ml} - \overline{p}_{ml}(t))^2 + \nu^p(t)\right) \\ &+ |\Omega|\log\sqrt{2\pi\nu^w}. \end{aligned} \tag{4.120}$$

To derive (4.120), one can start with the first term in (4.110) and leverage the Gaussianity of the approximated posterior on $\mathsf{x}_{nl}$:

$$\begin{aligned}&\sum_{n,l} D\Big(p_{\mathsf{x}_{nl}|\mathsf{r}_{nl}}(\,\cdot\,|\,\widehat{r}_{nl}(t);\nu^r_{nl}(t))\Big\|\,p_{\mathsf{x}_{nl}}(\cdot)\Big) \\ &= \sum_{n,l}\int_{x_{nl}}\mathcal{N}(x_{nl};\widehat{x}_{nl}(t),\nu^x_{nl}(t))\log\frac{\mathcal{N}(x_{nl};\widehat{x}_{nl}(t),\nu^x_{nl}(t))}{\mathcal{N}(x_{nl};\widehat{x}_0,\nu^x_0)},\end{aligned} \tag{4.121}$$

which then directly yields the first term in (4.120). The second term in (4.120) follows using a similar procedure, and the third and fourth terms follow directly from the PIAWGN model.

In the noise free setting (i.e., $\nu^w \to 0$), the third term in (4.120) dominates, avoiding the need to compute the other terms.

### 4.6.4 EM-BiG-AMP

For the likelihood (4.119) and priors (4.117)-(4.118), the distributional parameters $\boldsymbol{\theta} = [\nu^w, \widehat{x}_0, \nu_0^x]^T$ can be tuned using the EM approach from Section 4.5.1.[21] To initialize $\boldsymbol{\theta}$ for EM-BiG-AMP, we adapt the procedure outlined in [62] to our matrix-completion problem, giving the EM initializations $\widehat{x}_0 = 0$ and

$$\nu^w = \frac{\|P_\Omega(\boldsymbol{Y})\|_F^2}{(\mathrm{SNR}^0 + 1)|\Omega|} \tag{4.122}$$

$$\nu_0^x = \frac{1}{N}\left[\frac{\|P_\Omega(\boldsymbol{Y})\|_F^2}{|\Omega|} - \nu^w\right], \tag{4.123}$$

where $\mathrm{SNR}^0$ is an initial estimate of the signal-to-noise ratio that, in the absence of other knowledge, can be set at 100.

### 4.6.5 Rank selection

For MC rank-selection under the penalized log-likelihood strategy (4.115), we recommend using the *small sample corrected AIC* (AICc) [77] penalty $\eta(N) = 2\frac{|\Omega|}{|\Omega|-N_{\mathrm{eff}}-1}N_{\mathrm{eff}}$. For the MC problem, $N_{\mathrm{eff}} = \mathsf{df} + 3$, where $\mathsf{df} \triangleq N(M+L-N)$ counts the degrees-of-freedom in a rank-$N$ real-valued $M \times L$ matrix [78] and the three additional parameters come from $\boldsymbol{\theta}$. Based on the PIAWGN likelihood (4.119)

[21] For the first EM iteration, we recommend initializing BiG-AMP using $\nu_{nl}^x(1) = \nu_0^x$, $\widehat{x}_{nl}(1) = \widehat{x}_0$, $\nu_{mn}^a(1) = 1$, and $\widehat{a}_{mn}(1)$ drawn randomly from $p_{\mathsf{a}_{mn}}$. After the first iteration, we recommend warm-starting BiG-AMP using the values from the previous EM iteration.

and the standard form of the ML estimate of $\nu^w$ (see, e.g., [77, eq. (7)]), the update rule (4.115) becomes

$$\widehat{N} = \underset{N=1,\ldots,\overline{N}}{\arg\max} \left[ -|\Omega| \log \left( \frac{1}{|\Omega|} \sum_{(m,l)\in\Omega} \left( y_{ml} - \widehat{z}_{ml}(t) \right)^2 \right) \right.$$
$$\left. -2\frac{|\Omega|(N(M+L-N)+3)}{|\Omega| - N(M+L-N) - 4} \right]. \tag{4.124}$$

We note that a similar rule (but based on BIC rather than AICc) was used for rank-selection in [44].

MC rank selection can also be performed using the rank contraction scheme described in Section 4.5.2. We recommend choosing the maximum rank $\overline{N}$ to be the largest value such that $\overline{N}(M+L-\overline{N}) < |\Omega|$ and setting $\tau_{\text{MOS}} = 1.5$. Since the first EM iteration runs BiG-AMP with the large value $N = \overline{N}$, we suggest limiting the number of allowed BiG-AMP iterations during this first EM iteration to `nitFirstEM` $= 50$. In many cases, the rank learning procedure will correctly reduce the rank after these first few iterations, reducing the added computational cost of the rank selection procedure.

### 4.6.6 Matrix Completion Experiments

We now present the results of experiments used to ascertain the performance of BiG-AMP relative to existing state-of-the-art algorithms for matrix completion. For these experiments, we considered IALM [38], a nuclear-norm based convex-optimization method; LMaFit [42], a non-convex optimization-based approach using non-linear successive over-relaxation; GROUSE [46], which performs gradient descent on the Grassmanian manifold; Matrix-ALPS [43], a greedy hard-thresholding approach; and VSBL [53], a variational Bayes approach. In general, we configured

BiG-AMP as described in Section 4.6[22] and made our best attempt to configure the competing algorithms for maximum performance. That said, the different experiments that we ran required somewhat different parameter settings, as we detail in the sequel.

**Low-rank matrices**

We first investigate recovery of rank-$N$ matrices $\boldsymbol{Z} \in \mathbb{R}^{M \times L}$ from noiseless incomplete observations $\{z_{ml}\}_{(m,l)\in\Omega}$ with indices $\Omega$ chosen uniformly at random. To do this, we evaluated the normalized mean square error (NMSE) $\frac{\|\boldsymbol{Z}-\widehat{\boldsymbol{Z}}\|_F^2}{\|\boldsymbol{Z}\|_F^2}$ of the estimate $\widehat{\boldsymbol{Z}}$ returned by the various algorithms under test, examining 10 realizations of $(\boldsymbol{Z}, \Omega)$ at each problem size $(M, L, N)$. Here, each realization of $\boldsymbol{Z}$ was constructed as $\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{X}$ for $\boldsymbol{A}$ and $\boldsymbol{X}$ with iid $\mathcal{N}(0,1)$ elements.[23] All algorithms were forced[24] to use the true rank $N$, run under default settings with very minor modifications,[25] and terminated when the normalized change in either $\widehat{\boldsymbol{Z}}$ or $P_\Omega(\widehat{\boldsymbol{Z}})$ across iterations fell below the tolerance value of $10^{-8}$.

Defining "successful" matrix completion as NMSE $< -100$ dB, Fig. 4.2 shows the success rate of each algorithm over a grid of sampling ratios $\delta \triangleq \frac{|\Omega|}{ML}$ and ranks $N$.

[22]Unless otherwise noted, we used the BiG-AMP parameters $T_{\max} = 1500$ (see Section 4.2.8 for descriptions) and the adaptive damping parameters `stepInc` $= 1.1$, `stepDec` $= 0.5$, `stepMin` $= 0.05$, `stepMax` $= 0.5$, `stepWindow` $= 1$, and $\beta(1) =$ `stepMin`. (See Section 4.4.2 for descriptions).

[23]We chose the i.i.d Gaussian construction due to its frequent appearance in the matrix-completion literature. Similar performance was observed when the low-rank factors $\boldsymbol{A}$ and $\boldsymbol{X}$ were generated in other ways, such as from the left and right singular vectors of an i.i.d Gaussian matrix.

[24]This restriction always improved the performance of the tested algorithms.

[25]GROUSE was run with `maxCycles` $= 600$ and `step_size` $= 0.5$, where the latter was chosen as a good compromise between phase-transition performance and runtime. VSBL was run under `MAXITER` $= 2000$ and fixed $\beta = 10^9$; adaptive selection of $\beta$ was found to produce a significant degradation in the observed phase transition. LMaFit was run from the same random initialization as BiG-AMP and permitted at most `maxit` $= 6000$ iterations. IALM was allowed at most 2000 iterations. A maximum runtime of one hour per realization was enforced for all algorithms.

As a reference, the solid line superimposed on each subplot delineates the problem feasibility boundary, i.e., the values of $(\delta, N)$ yielding $|\Omega| = \mathsf{df}$, where $\mathsf{df} = N(M + L - N)$ is the degrees-of-freedom in a rank-$N$ real-valued $M \times L$ matrix; successful recovery above this line is impossible by any method.

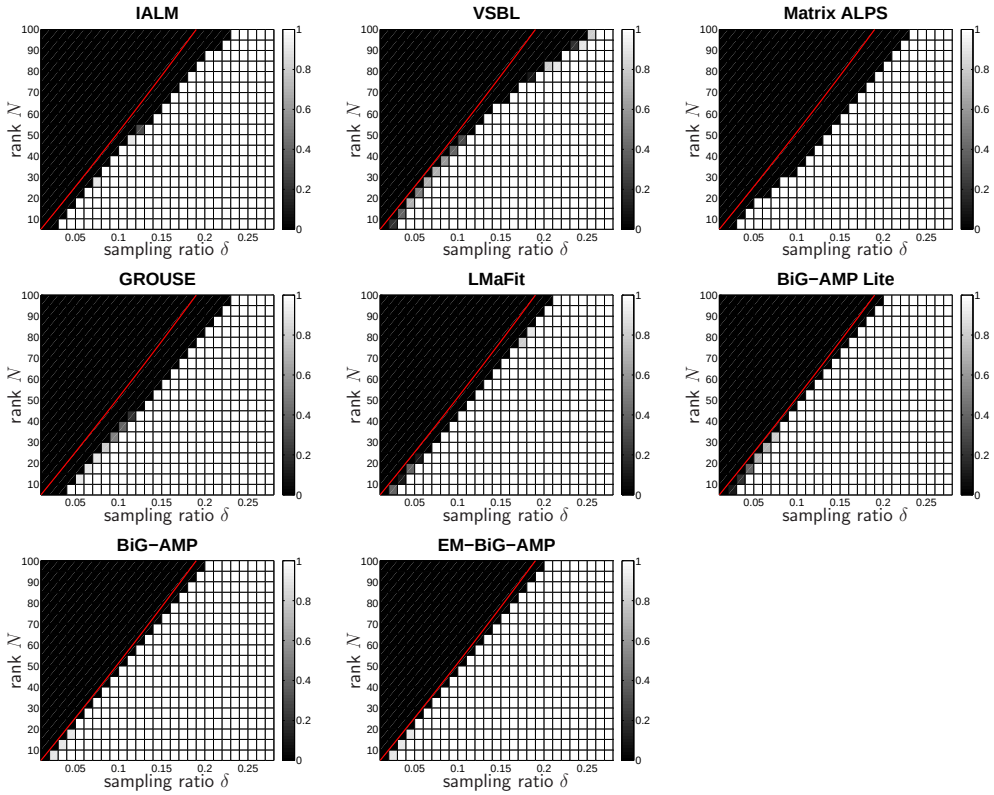


Figure 4.2: Empirical success rates for noiseless completion of an $M \times L$ matrix sampled uniformly at random, as a function of sampling ratio $\delta = \frac{|\Omega|}{ML}$ and rank $N$. Here, "success" is defined as NMSE $< -100$ dB, success rates were computed from 10 random realizations, and $M = L = 1000$. Points above the red curve are infeasible, as described in the text.

Figure 4.2 shows that each algorithm exhibits a sharp phase-transition separating near-certain success from near-certain failure. There we see that BiG-AMP yields the best PTC. Moreover, BiG-AMP's PTC is *near optimal* in the sense of coming

very close to the feasibility boundary for all tested $\delta$ and $N$. In addition, Fig. 4.2 shows that BiG-AMP-Lite yields the second-best PTC, which matches that of BiG-AMP except under very low sampling rates (e.g., $\delta < 0.03$). Recall that the only difference between the two algorithms is that BiG-AMP-Lite uses the scalar-variance simplification from Section 4.3.1.

Figure 4.3 plots median runtime[26] to NMSE $= -100$ dB versus rank $N$ for several sampling ratios $\delta$, uncovering orders-of-magnitude differences among algorithms. For most values of $\delta$ and $N$, LMaFit was the fastest algorithm and BiG-AMP-Lite was the second fastest, although BiG-AMP-Lite was faster than LMaFit at small $\delta$ and relatively large $N$, while BiG-AMP-Lite was slower than GROUSE at large $\delta$ and very small $N$. In all cases, BiG-AMP-Lite was faster than IALM and VSBL, with several orders-of-magnitude difference at high rank. Meanwhile, EM-BiG-AMP was about 3 to 5 times slower than BiG-AMP-Lite. Although none of the algorithm implementations were fully optimized, we believe that the reported runtimes are insightful, especially with regard to the scaling of runtime with rank $N$.

**Approximately low-rank matrices**

Next we evaluate the performance of recovering approximately low-rank matrices by repeating an experiment from the LMaFit paper [42]. For this, we constructed the complete matrix as $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{T}} \in \mathbb{R}^{500\times 500}$, where $\boldsymbol{U}, \boldsymbol{V}$ were orthogonal matrices (built by orthogonalizing iid $\mathcal{N}(0,1)$ matrices using MATLAB's `orth` command) and $\boldsymbol{\Sigma}$ was a positive diagonal matrix containing the singular values of $\boldsymbol{Z}$. Two versions of

[26]The reported runtimes do not include the computations used for initialization nor those used for runtime evaluation.

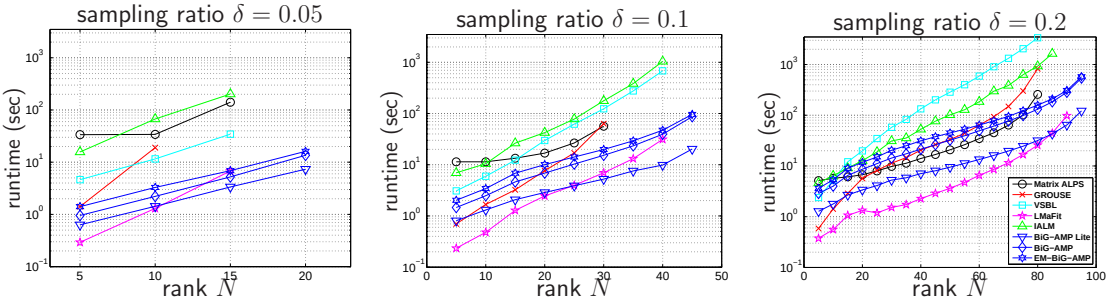


Figure 4.3: Runtime to NMSE= $-100$ dB for noiseless completion of an $M \times L$ matrix sampled uniformly at random, versus rank $N$, at $M = L = 1000$ and several sampling ratios $\delta = \frac{|\Omega|}{ML}$. All results represent median performance over 10 trials. Missing values indicate that the algorithm did not achieve the required NMSE before termination and correspond to the black regions in Fig. 4.2.

$\mathbf{\Sigma}$ were considered: one with exponentially decaying singular values $[\mathbf{\Sigma}]_{m,m} = e^{-0.3m}$, and one with the power-law decay $[\mathbf{\Sigma}]_{m,m} = m^{-3}$.

As in [42], we first tried to recover $\boldsymbol{Z}$ from the noiseless incomplete observations $\{z_{ml}\}_{(m,l)\in\Omega}$, with $\Omega$ chosen uniformly at random. Figure 4.4 shows the performance of several algorithms that are able to learn the underlying rank: LMaFit,[27] VSBL,[28] and EM-BiG-AMP under the penalized log-likelihood rank selection strategy from Section 4.5.2.[29] All three algorithms were allowed a maximum rank of $\overline{N} = 30$. The figure shows that the NMSE performance of BiG-AMP and LMaFit are similar, although BiG-AMP tends to find solutions with lower rank but comparable NMSE at low sampling ratios $\delta$. For this noiseless experiment, VSBL consistently estimates ranks that are too low, leading to inferior NMSEs.

[27]LMaFit was run under the settings provided in their source code for this example.

[28]VSBL was allowed at most 100 iterations and run with `DIMRED_THR` $= 10^3$, `UPDATE_BETA` $= 1$, and tolerance $= 10^{-8}$.

[29]Rank-selection rule (4.115) was used with up to 5 EM iterations for each rank hypothesis $N$, a minimum of 30 and maximum of 100 BiG-AMP iterations for each EM iteration, and a BiG-AMP tolerance of $10^{-8}$.

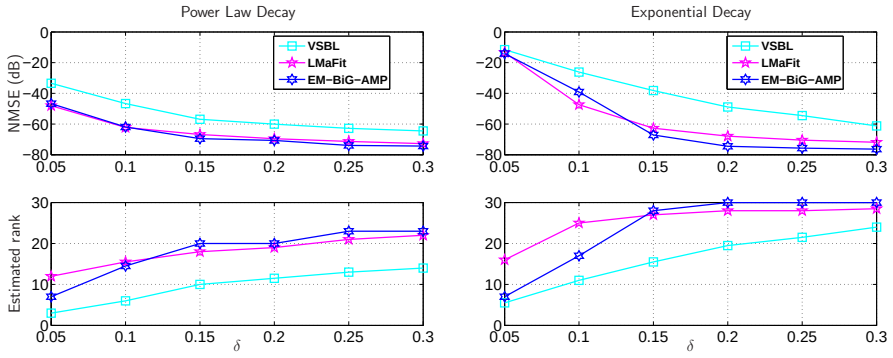


Figure 4.4: NMSE (top) and estimated rank (bottom) in noiseless completion of an $M \times L$ matrix sampled uniformly at random, versus sampling ratio $\delta = \frac{|\Omega|}{ML}$. The complete matrices were approximately low-rank, with power-law (left) and exponential (right) singular-value decays and $M = L = 500$. All results represent median performance over 10 random trials.

Next, we examined *noisy* matrix completion by constructing the matrix $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{T}}$ as above but then corrupting the measurements with AWGN. Figure 4.5 shows NMSE and estimated rank versus the measurement signal-to-noise ratio (SNR) $\sum_{(m,l)\in\Omega} |z_{ml}|^2 / \sum_{(m,l)\in\Omega} |y_{ml} - z_{ml}|^2$ at a sampling rate of $\delta = 0.2$. There we see that, for SNRs $< 50$ dB, EM-BiG-AMP and VSBL offer similarly good NMSE performance and nearly identical rank estimates, whereas LMaFit overestimates the rank and thus performs worse in NMSE. Meanwhile, for SNRs $> 50$ dB, EM-BiG-AMP and LMaFit offer similarly good NMSE performance and nearly identical rank estimates, whereas VSBL underestimates the rank and thus performs worse in NMSE. Thus, in these examples, EM-BiG-AMP is the only algorithm to successfully estimate the rank across the full SNR range.

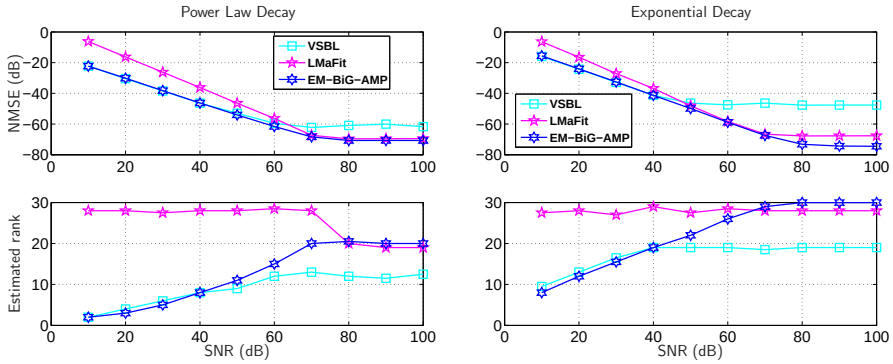


Figure 4.5: NMSE (top) and estimated rank (bottom), versus SNR, in noisy completion of an $500 \times 500$ matrix sampled uniformly at random at rate $\delta = 0.2$. The complete matrices were approximately low-rank, with power-law (left) and exponential (right) singular-value decays.

### Image completion

We now compare the performance of several matrix-completion algorithms for the task of reconstructing an image from a subset of its pixels. For this, we repeated the experiment in the Matrix-ALPS paper [43], where the $512 \times 512$ boat image was reconstructed from 35% of its pixels sampled uniformly at random. Figure 4.6 shows the complete (full-rank) image, the images reconstructed by several matrix-completion algorithms[30] under a fixed rank of $N = 40$, and the NMSE-minimizing rank-40 approximation of the complete image, computed using an SVD. In all cases, the sample mean of the observations was subtracted prior to processing and then added back to the estimated images, since this approach generally improved performance. Figure 4.6 also lists the median reconstruction NMSE over 10 sampling-index realizations

[30] All algorithms were run with a convergence tolerance of $10^{-4}$. VSBL was run with $\beta$ hand-tuned to maximize performance, as the adaptive version did not converge on this example. GROUSE was run with `maxCycles` $= 600$ and `step_size` $= 0.1$. Matrix-ALPS II with QR was run under default parameters and 300 allowed iterations. Other settings are similar to earlier experiments.

$\Omega$. From these results, it is apparent that EM-BiG-AMP provides the best NMSE, which is only 3 dB from that of the NMSE-optimal rank-40 approximation.

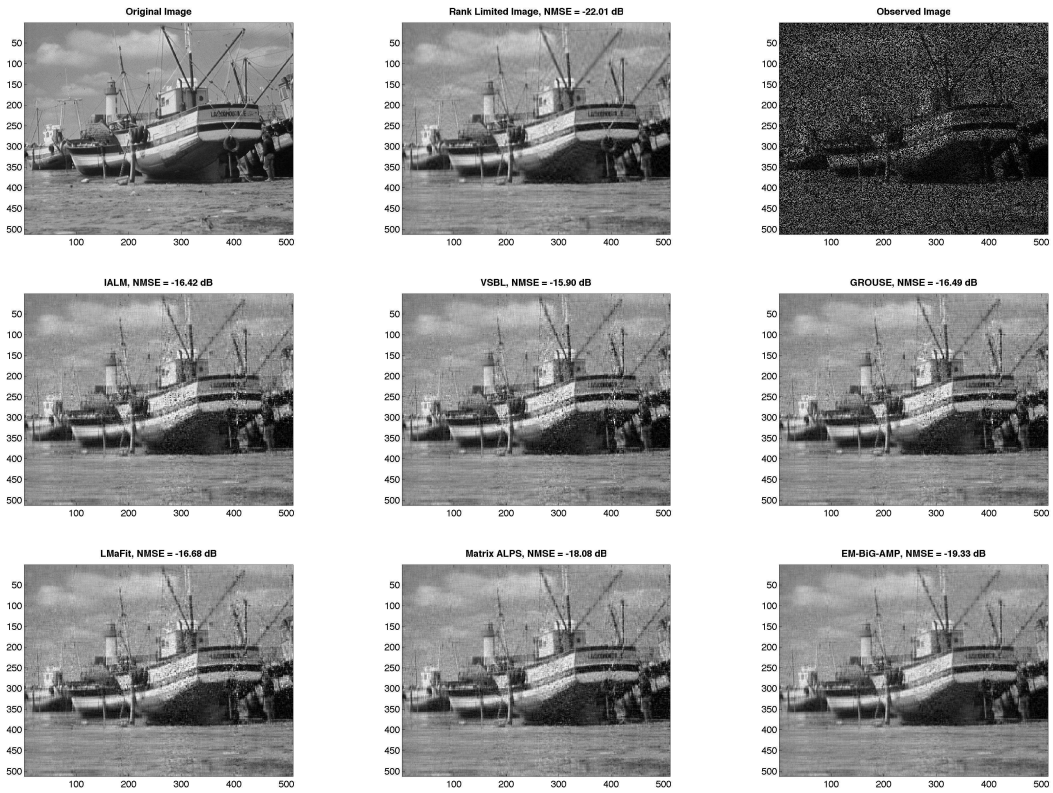


Figure 4.6: For the image completion experiment, the complete image is shown on the top left, its best rank-40 approximation is shown on the top middle, and the observed image with 35% of the pixels observed is shown on the top right. The other panes show various algorithms' image reconstructions from 35% of the complete-image pixels (selected uniformly at random) as well as the mean NMSE over 10 trials.

### Collaborative Filtering

In our final experiment, we investigate the performance of several matrix-completion algorithms on the task of collaborative filtering. For this, we repeated an experiment from the VSBL paper [53] that used the MovieLens 100k dataset,

which contains ratings $\{z_{ml}\}_{(m,l)\in\mathcal{R}}$, where $|\mathcal{R}| = 100\,000$ and $z_{ml} \in \{1,2,3,4,5\}$, from $M = 943$ users about $L = 1682$ movies. The algorithms were provided with a randomly chosen training subset $\{z_{ml}\}_{(m,l)\in\Omega}$ of the ratings (i.e., $\Omega \subset \mathcal{R}$) from which they estimated the unseen ratings $\{\widehat{z}_{ml}\}_{(m,l)\in\mathcal{R}\setminus\Omega}$. Performance was then assessed by computing the Normalized Mean Absolute Error (NMAE)

$$\text{NMAE} = \frac{1}{4|\mathcal{R} \setminus \Omega|} \sum_{(m,l)\in\mathcal{R}\setminus\Omega} |z_{ml} - \widehat{z}_{ml}|, \tag{4.125}$$

where the 4 in the denominator of (4.125) reflects the difference between the largest and smallest user ratings (i.e., 5 and 1). When constructing $\Omega$, we used a fixed percentage of the ratings given by each user and made sure that at least one rating was provided for every movie in the training set.

Figure 4.7 reports the NMAE and estimated rank $\widehat{N}$ for EM-BiG-AMP under the PIAWGN model (4.119), LMaFit, and VSBL,[31] all of which include mechanisms for rank estimation. Figure 4.7 shows that, under the PIAWGN model, EM-BiG-AMP yields NMAEs that are very close to those of VSBL[32] but slightly inferior at larger training fractions, whereas LMaFit returns NMAEs that are substantially worse all training fractions.[33] Figure 4.7 also shows that LMaFit's estimated rank is much higher than those of VSBL and EM-BiG-AMP, suggesting that its poor NMAE performance is the result of overfitting. (Recall that similar behavior was seen for

[31]VSBL was was allowed at most 100 iterations and was run with `DIMRED_THR`$= 10^3$ and `UPDATE_BETA`$= 1$. Both VSBL and EM-BiG-AMP used a tolerance of $10^{-8}$. LMaFit was configured as for the MovieLens experiment in [42]. Each algorithm was allowed a maximum rank of $\overline{N} = 30$.

[32]The NMAE values reported for VSBL in Fig. 4.7 are slightly inferior to those reported in [53]. We attribute the discrepancy to differences in experimental setup, such as the construction of $\Omega$.

[33]The NMAE results presented here differ markedly from those in the MovieLens experiment in [42] because, in the latter paper, the entire set of ratings was used for both training *and testing*, with the (trivial) result that high-rank models (e.g., $\widehat{N} = 94$) yield nearly zero test error.

noisy matrix completion in Fig. 4.5.) In addition, Fig. 4.7 shows that, as the training fraction increases, EM-BiG-AMP's estimated rank remains very low (i.e., $\leq 2$) while that of VSBL steady increases (to $> 10$). This prompts the question: is VSBL's excellent NMAE the result of accurate rank estimation or the use of a heavy-tailed (i.e., student's t) noise prior?

To investigate the latter question, we ran BiG-AMP under

$$p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml} \,|\, z_{ml}) = \begin{cases} \frac{\lambda}{2}\exp\big(-\lambda|y_{ml}-z_{ml}|\big) & (m,l) \in \Omega \\ 1_{y_{ml}} & (m,l) \notin \Omega, \end{cases} \tag{4.126}$$

i.e., a possibly incomplete additive white Laplacian noise (PIAWLN) model, and used the EM-based approach from Section 4.5.1 to learn the rate parameter $\lambda$. Figure 4.7 shows that, under the PIAWLN model, EM-BiG-AMP essentially matches the NMAE performance of VSBL and even improves on it at very low training fractions. Meanwhile, its estimated rank $\widehat{N}$ remains low for all training fractions, suggesting that the use of a heavy-tailed noise model was the key to achieving low NMAE in this experiment. Fortunately, the generality and modularity of BiG-AMP made this an easy task.

**Summary**

In summary, the known-rank synthetic-data results above showed the EM-BiG-AMP methods yielding phase-transition curves superior to all other algorithms under test. In addition, they showed BiG-AMP-Lite to be the second fastest algorithm (behind LMaFit) for most combinations of sampling ratio $\delta$ and rank $N$, although it was the fastest for small $\delta$ and relatively high $N$. Also, they showed EM-BiG-AMP was about 3 to 5 times slower than BiG-AMP-Lite but still much faster than IALM and VSBL at high ranks. Meanwhile, the unknown-rank synthetic-data results

above showed EM-BiG-AMP yielding excellent NMSE performance in both noiseless and noisy scenarios. For example, in the noisy experiment, EM-BiG-AMP uniformly outperformed its competitors (LMaFit and VSBL).

In the image completion experiment, EM-BiG-AMP again outperformed all competitors, beating the second best algorithm (Matrix ALPS) by more than 1 dB and the third best algorithm (LMaFit) by more than 2.5 dB. Finally, in the collaborative filtering experiment, EM-BiG-AMP (with the PIAWLN likelihood model) matched the best competitor (VSBL) in terms of NMAE, and significantly outperformed the second best (LMaFit).

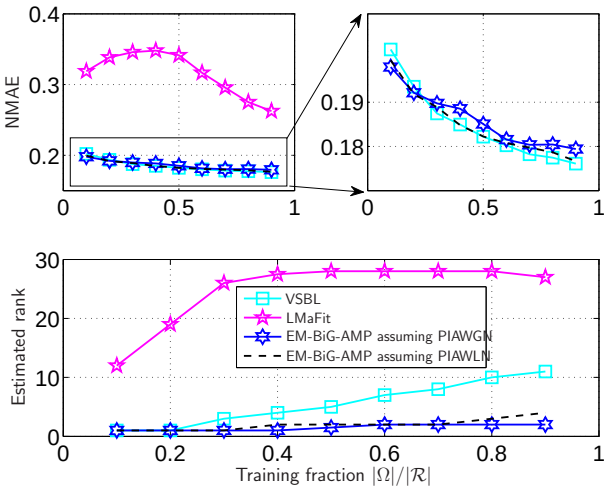


Figure 4.7: Median NMAE (top) and estimated rank (bottom) for movie-rating prediction versus fraction of training data $|\Omega|/|\mathcal{R}|$ over 10 trials for the 100k MovieLens data set.

## 4.7 Robust PCA

### 4.7.1 Problem Setup

In robust principal components analysis (RPCA) [82], one seeks to estimate a low-rank matrix observed in the presence of noise and large outliers. The data model for RPCA can be written as

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{E} + \boldsymbol{W}, \tag{4.127}$$

where $\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{X}$—the product of tall $\boldsymbol{A}$ and wide $\boldsymbol{X}$—is the low-rank matrix of interest, $\boldsymbol{E}$ is a *sparse* outlier matrix, and $\boldsymbol{W}$ is a dense noise matrix. We now suggest two ways of applying BiG-AMP to the RPCA problem, both of which treat the elements of $\boldsymbol{A}$ as iid $\mathcal{N}(0, \nu_0^a)$ similar to (4.117), the elements of $\boldsymbol{X}$ as iid $\mathcal{N}(0, \nu_0^x)$ similar to (4.118), the *non-zero* elements of $\boldsymbol{E}$ as iid $\mathcal{N}(0, \nu_1)$, and the elements of $\boldsymbol{W}$ as iid $\mathcal{N}(0, \nu_0)$, with $\nu_1 \gg \nu_0$.

In the first approach, $\boldsymbol{E} + \boldsymbol{W}$ is treated as additive noise on $\boldsymbol{Z}$, leading to the likelihood model

$$\begin{aligned} &p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml} \,|\, z_{ml}) \\ &\quad = (1-\lambda)\mathcal{N}(y_{ml}; z_{ml}, \nu_0) + \lambda\mathcal{N}(y_{ml}; z_{ml}, \nu_0 + \nu_1), \end{aligned} \tag{4.128}$$

where $\lambda \in [0, 1]$ models outlier density.

In the second approach, $\boldsymbol{W}$ is treated as additive noise but $\boldsymbol{E}$ is treated as an additional estimand. In this case, by multiplying both sides of (4.127) by any (known) unitary matrix $\boldsymbol{Q} \in \mathbb{R}^{M \times M}$, we can write

$$\underbrace{\boldsymbol{Q}\boldsymbol{Y}}_{\triangleq \underline{\boldsymbol{Y}}} = \underbrace{\begin{bmatrix} \boldsymbol{Q}\boldsymbol{A} & \boldsymbol{Q} \end{bmatrix}}_{\triangleq \underline{\boldsymbol{A}}} \underbrace{\begin{bmatrix} \boldsymbol{X} \\ \boldsymbol{E} \end{bmatrix}}_{\triangleq \underline{\boldsymbol{X}}} + \underbrace{\boldsymbol{Q}\boldsymbol{W}}_{\triangleq \underline{\boldsymbol{W}}}, \tag{4.129}$$

and apply BiG-AMP to the "augmented" model $\underline{\underline{\boldsymbol{Y}}} = \underline{\underline{\boldsymbol{A}}}\underline{\underline{\boldsymbol{X}}} + \underline{\underline{\boldsymbol{W}}}$. Here, $\underline{\underline{\boldsymbol{W}}}$ remains iid $\mathcal{N}(0, \nu_0)$, thus giving the likelihood

$$p_{\underline{\mathsf{y}}_{ml}|\underline{\mathsf{z}}_{ml}}(\underline{y}_{ml} \,|\, \underline{z}_{ml}) = \mathcal{N}(\underline{y}_{ml}; \underline{z}_{ml}, \nu_0). \tag{4.130}$$

Meanwhile, we choose the following separable priors on $\underline{\underline{\boldsymbol{A}}}$ and $\underline{\underline{\boldsymbol{X}}}$:

$$p_{\underline{\mathsf{a}}_{mn}}(\underline{a}_{mn}) = \begin{cases} \mathcal{N}(\underline{a}_{mn}; 0, \nu_0^a) & n \le N \\ \mathcal{N}(\underline{a}_{mn}; q_{mn}, 0) & n > N \end{cases} \tag{4.131}$$

$$p_{\underline{\mathsf{x}}_{nl}}(\underline{x}_{nl}) = \begin{cases} \mathcal{N}(\underline{x}_{nl}; 0, \nu_0^x) & n \le N \\ (1-\lambda)\mathtt{1}_{\underline{x}_{nl}} + \lambda\mathcal{N}(\underline{x}_{nl}; 0, \nu_1) & n > N. \end{cases} \tag{4.132}$$

Essentially, the first $N$ columns of $\underline{\underline{\boldsymbol{A}}}$ and first $N$ rows of $\underline{\underline{\boldsymbol{X}}}$ model the factors of the low-rank matrix $\boldsymbol{AX}$, and thus their elements are assigned iid Gaussian priors, similar to (4.117)-(4.118) in the case of matrix completion. Meanwhile, the last $M$ rows in $\underline{\underline{\boldsymbol{X}}}$ are used to represent the sparse outlier matrix $\boldsymbol{E}$, and thus their elements are assigned a Bernoulli-Gaussian prior. Finally, the last $M$ columns of $\underline{\underline{\boldsymbol{A}}}$ are used to represent the designed matrix $\boldsymbol{Q}$, and thus their elements are assigned zero-variance priors. Since we find that BiG-AMP is numerically more stable when $\boldsymbol{Q}$ is chosen as a dense matrix, we set it equal to the singular-vector matrix of an iid $\mathcal{N}(0,1)$ matrix. After running BiG-AMP, we can recover an estimate of $\boldsymbol{A}$ by left multiplying the estimate of $\underline{\underline{\boldsymbol{A}}}$ by $\boldsymbol{Q}^{\mathsf{H}}$.

### 4.7.2 Initialization

We recommend initializing $\widehat{\underline{a}}_{mn}(1)$ using a random draw from its prior and initializing $\widehat{\underline{x}}_{nl}(1)$ at the mean of its prior, i.e., $\widehat{\underline{x}}_{nl}(1) = 0$. The latter tends to perform better than initializing $\widehat{\underline{x}}_{nl}(1)$ randomly, because it allows the measurements $\boldsymbol{Y}$ to determine the initial locations of the outliers in $\boldsymbol{E}$. As in Section 4.6.2, we suggest

initializing $\nu^a_{mn}(1)$ and $\nu^x_{nl}(1)$ at 10 times the variance of their respective priors to emphasize the role of the measurements during the first few iterations.

### 4.7.3 EM-BiG-AMP

The EM approach from Section 4.5.1 can be straightforwardly applied to BiG-AMP for RPCA: after fixing $\nu^a_0 = 1$, EM can be used to tune the remaining distributional parameters, $\boldsymbol{\theta} = [\nu_0, \nu_1, \nu^x_0, \lambda]^T$. To avoid initializing $\nu_0$ and $\nu^x_0$ with overly large values in the presence of large outliers $e_{nl}$, we suggest the following procedure. First, define the set $\Gamma \triangleq \big\{(m,l) : |y_{ml}| \leq \text{median}\{|y_{ml}|\}\big\}$ and its complement $\Gamma^c$. Then initialize

$$\nu_0 = \frac{\frac{1}{|\Gamma|}\sum_{(m,l)\in\Gamma}|y_{ml}|^2}{\text{SNR}^0 + 1} \tag{4.133}$$

$$\nu^x_0 = \frac{1}{N}\text{SNR}^0\nu_0 \tag{4.134}$$

$$\nu_1 = \frac{1}{|\Gamma^c|}\sum_{(m,l)\in\Gamma^c}|y_{ml}|^2, \tag{4.135}$$

where, as in Section 4.6.4, we suggest setting $\text{SNR}^0 = 100$ in the absence of prior knowledge. This approach uses the median to avoid including outliers among the samples used to estimate the variances of the dense-noise and low-rank components. Under these rules, the initialization $\lambda = 0.1$ was found to work well for most problems.

### 4.7.4 Rank Selection

In many applications of RPCA, such as video separation, the singular-value profile of $\boldsymbol{AX}$ exhibits a sharp cutoff, in which case it is recommended to perform rank-selection using the contraction strategy from Section 4.5.2.

### 4.7.5 Avoiding Local Minima

Sometimes, when $N$ is very small, BiG-AMP may converge to a local solution that mistakes entire rows or columns of $\boldsymbol{AX}$ for outliers. Fortunately, this situation is easy to remedy with a simple heuristic procedure: the posterior probability that $y_{ml}$ is outlier-corrupted can be computed for each $(m, l)$ at convergence, and if any of the row-wise sums exceeds $0.8M$ or any of the column-wise sums exceeds $0.8L$, then BiG-AMP is restarted from a new random initialization. Experimentally, we found that one or two of such restarts is generally sufficient to avoid local minima.

### 4.7.6 Robust PCA Experiments

In this section, we present a numerical study of the two BiG-AMP formulations of RPCA proposed in Section 4.7, including a comparison to the state-of-the-art IALM [38], LMaFit [42], GRASTA [49], and VSBL [53] algorithms. In the sequel, we use "BiG-AMP-1" when referring to the formulation that treats the outliers as noise, and "BiG-AMP-2" when referring to the formulation that explicitly estimates the outliers.

#### Phase Transition Behavior

We first study the behavior of the proposed BiG-AMP algorithms for RPCA on noise-free synthetic problems. For this, we generated problem realizations of the form $\boldsymbol{Y} = \boldsymbol{Z} + \boldsymbol{E}$, where the low-rank component $\boldsymbol{Z} = \boldsymbol{AX}$ was generated from $\boldsymbol{A}$ and $\boldsymbol{X}$ with iid $\mathcal{N}(0, 1)$ entries, and where the sparse corruption matrix $\boldsymbol{E}$ had a fraction $\delta$ of non-zero entries that were located uniformly at random with amplitudes drawn iid uniform on $[-10, 10]$. The dimensions of $\boldsymbol{Y} \in \mathbb{R}^{M \times L}$ were fixed at $M = L = 200$, the rank $N$ (of $\boldsymbol{Z}$) was varied from 10 to 90, and the outlier fraction $\delta$ was varied from

0.05 to 0.45. We note that, under these settings, the outlier magnitudes are on the same order as the magnitudes of $\boldsymbol{Z}$, which is the most challenging case: much larger outliers would be easier to detect, after which the corrupted elements of $\boldsymbol{Y}$ could be safely treated as incomplete, whereas much smaller outliers could be treated like AWGN.

All algorithms under test were run to a convergence tolerance of $10^{-8}$ and forced to use the true rank $N$. GRASTA, LMaFit, and VSBL were run under their recommended settings.[34] Two versions of IALM were tested: "IALM-1," which uses the universal penalty parameter $\lambda_{\texttt{ALM}} = \frac{1}{\sqrt{M}}$, and "IALM-2," which tries 50 hypotheses of $\lambda_{\texttt{ALM}}$, logarithmically spaced from $\frac{1}{10\sqrt{M}}$ to $\frac{10}{\sqrt{M}}$ and uses an oracle to choose the MSE-minimizing hypothesis. BiG-AMP-1 and BiG-AMP-2 were given perfect knowledge of the mean and variance of the entries of $\boldsymbol{A}$, $\boldsymbol{X}$, and $\boldsymbol{E}$ (although their Bernoulli-Gaussian model of $\boldsymbol{E}$ did not match the data generation process) as well as the outlier density $\lambda$, while EM-BiG-AMP-2 learned all model parameters from the data. BiG-AMP-1 was run under a fixed damping of $\beta = 0.25$, while BiG-AMP-2 was run under adaptive damping with $\texttt{stepMin} = 0.05$ and $\texttt{stepMax} = 0.5$. Both variants used a maximum of 5 restarts to avoid local minima.

Figure 4.8 shows the empirical success rate achieved by each algorithm as a function of corruption-rate $\delta$ and rank $N$, averaged over 10 trials, where a "success" was defined as attaining an NMSE of $-80$ dB or better in the estimation of the low-rank component $\boldsymbol{Z}$. The red curves in Fig. 4.8 delineate the problem feasibility boundary: for points $(\delta, N)$ above the curve, $N(M+L-N)$, the degrees-of-freedom in $\boldsymbol{Z}$, exceeds

[34]For LMaFit, however, we increased the maximum number of allowed iterations, since this improved its performance.

$(1-\delta)ML$, the number of uncorrupted observations, making it impossible to recover $\boldsymbol{Z}$ without additional information.

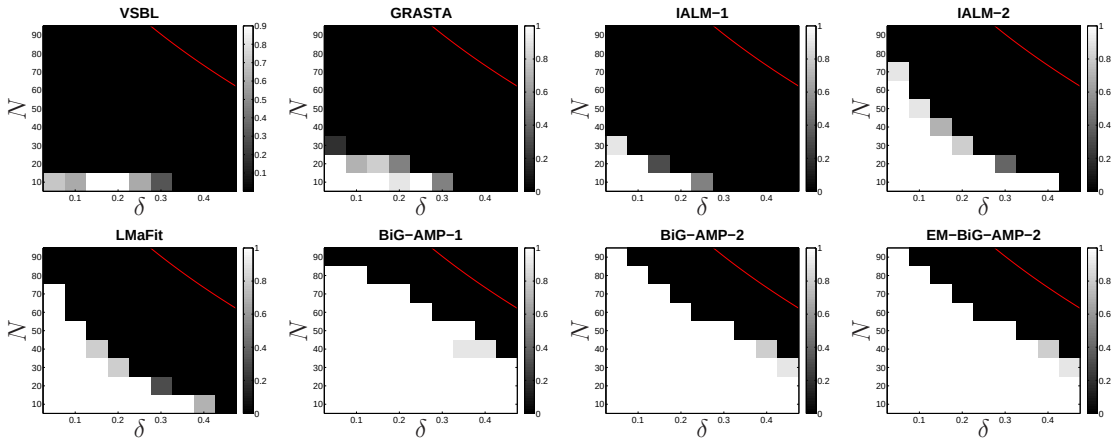


Figure 4.8: Empirical success rates for RPCA with a $200 \times 200$ matrix of rank $N$ corrupted by a fraction $\delta$ of outliers with amplitudes uniformly distributed on $[-10, 10]$. Here, "success" is defined as NMSE $< -80$ dB, and success rates were averaged over 10 problem realizations. Points above the red curve are infeasible, as described in the text.

Figure 4.8 shows that all algorithms exhibit a relatively sharp phase-transition curve (PTC) separating success and failure regions, and that the BiG-AMP algorithms achieve substantially better PTCs than the other algorithms. The PTCs of BiG-AMP-1 and BiG-AMP-2 are similar (but not identical), suggesting that both formulations are equally effective. Meanwhile, the PTCs of BiG-AMP-2 and EM-BiG-AMP-2 are nearly identical, demonstrating that the EM procedure was able to successfully learn the statistical model parameters used by BiG-AMP. Figure 4.8 also shows that all RPCA phase transitions remain relatively far from the feasibility boundary, unlike those for matrix completion (MC) shown in Fig. 4.2. This behavior, also observed in [82], is explained by the relative difficulty of RPCA over MC: the locations of

RPCA outliers (which in this case effectively render the corrupted observations as incomplete) are unknown, whereas in MC they are known.

Figure 4.9 plots runtime to NMSE $= -80$ dB as a function of rank $N$ for various outlier fractions. The results suggest that the BiG-AMP algorithms are moderate in terms of speed, being faster than GRASTA[35] and much faster than the grid-tuned IALM-2, but slower than IALM-1, VSBL, and LMaFit. Notably, among the non-BiG-AMP algorithms, LMaFit offers both the fastest runtime and the best phase-transition curve on this synthetic test problem.

In summary, the results presented here suggest that BiG-AMP achieves state-of-the-art PTCs while maintaining runtimes that are competitive with existing approaches.

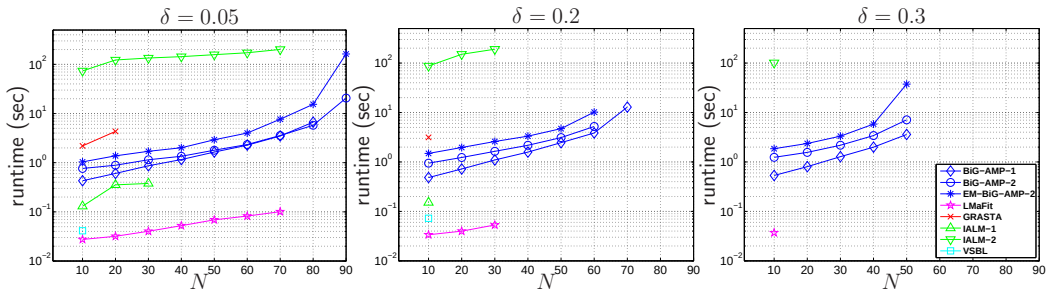


Figure 4.9: Runtime to NMSE$= -80$ dB for RPCA with a $200 \times 200$ matrix of rank $N$ corrupted by a fraction $\delta \in \{0.05, 0.2, 0.3\}$ of outliers with amplitudes uniformly distributed on $[-10, 10]$. All results represent median performance over 10 trials. Missing values indicate that the algorithm did not achieve the required NMSE before termination and correspond to the black regions in Fig. 4.8.

[35]We note that, for this experiment, GRASTA was run as a Matlab M-file and not a MEX file, because the MEX file would not compile on the supercomputer used for the numerical results. That said, since BiG-AMP was also run as an unoptimized M-file, the comparison could be considered "fair."

**Rank Estimation**

We now investigate the ability to estimate the underlying rank, $N$, for EM-BiG-AMP-2 (using the rank-contraction strategy from Section 4.5.2[36]) IALM-1, IALM-2, LMaFit, and VSBL, all of which include either explicit or implicit rank-selection mechanisms. For this, we generated problem realizations of the form $\boldsymbol{Y} = \boldsymbol{Z}+\boldsymbol{E}+\boldsymbol{W}$, where the $200 \times 200$ rank-$N$ matrix $\boldsymbol{Z}$ and $\delta = 0.1$-sparse outlier matrix $\boldsymbol{E}$ were generated as described in Section 4.7.6 and the noise matrix $\boldsymbol{W}$ was constructed with iid $\mathcal{N}(0, 10^{-3})$ elements. The algorithms under test were not provided with knowledge of the true rank $N$, which was varied between 5 and 90. LMaFit, VSBL, and EM-BiG-AMP, were given an initial rank estimate of $\overline{N} = 90$, which enforces an upper bound on the final estimates that they report.

Figure 4.10 reports RPCA performance versus (unknown) true rank $N$ in terms of the estimated rank $\widehat{N}$ and the NMSE on the estimate $\widehat{\boldsymbol{Z}}$. All results represent median performance over 10 Monte-Carlo trials. The figure shows that EM-BiG-AMP-2 and LMaFit returned accurate rank estimates $\widehat{N}$ over the full range of true rank $N \in [5, 90]$, whereas VSBL returned accurate rank estimates only for $N \leq 20$, and both IALM-1 and IALM-2 greatly overestimated the rank at all $N$. Meanwhile, Fig. 4.10 shows that EM-BiG-AMP-2 and LMaFit returned accurate estimates of $\widehat{\boldsymbol{Z}}$ for all $N \leq 80$ (with EM-BiG-AMP-2 outperforming LMaFit by several dB throughout this range), whereas VSBL and IALM-1 and IALM-2 returned accurate estimates of $\widehat{\boldsymbol{Z}}$ only for small values of $N$. We note that the relatively poor MSE performance of LMaFit and EM-BiG-AMP-2 for true rank $N > 80$ is not due to poor rank estimation

[36]The rank-selection rule (4.116) was used with $\tau_{\text{MOS}} = 5$, up to 50 EM iterations, and a minimum of 30 and maximum of 500 BiG-AMP iterations per EM iteration.

but rather due to the fact that, at $\delta = 0.1$, these operating points lie above the PTCs shown in Fig. 4.8.

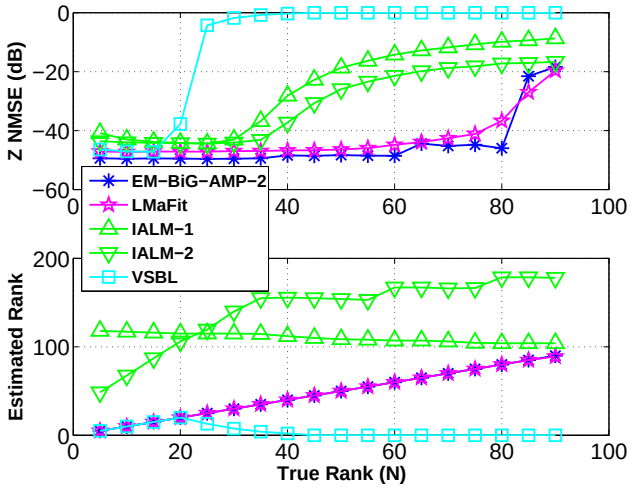


Figure 4.10: NMSE (top) and estimated rank $\widehat{N}$ (bottom) versus true rank $N$ for several algorithms performing RPCA on a $200 \times 200$ matrix in the presence of additive $\mathcal{N}(0, 10^{-3})$ noise and a fraction $\delta = 0.1$ of outliers with amplitudes uniformly distributed on $[-10, 10]$. All results represent the median over 10 trials.

### Application to Video Surveillance

We now apply EM-BiG-AMP-2 to a video surveillance problem, where the goal is to separate a video sequence into a static "background" component and a dynamic "foreground" component. To do this, we stack each frame of the video sequence into a single column of the matrix $\boldsymbol{Y}$, run EM-BiG-AMP-2 as described in Section 4.7, extract the background frames from the estimate of the low-rank component $\boldsymbol{Z} = \boldsymbol{AX}$, and extract the foreground frames from the estimate of the (sparse) outlier component $\boldsymbol{E}$. We note that a perfectly time-invariant background would correspond

to a rank-one $\boldsymbol{Z}$ and that the noise term $\boldsymbol{W}$ in (4.127) can be used to account for small perturbations that are neither low-rank nor sparse.

We tested EM-BiG-AMP[37] on the popular "mall" video sequence,[38] processing 200 frames (of $256 \times 320$ pixels each) using an initial rank estimate of $\overline{N} = 5$. Figure 4.11 shows the result, with original frames in the left column and EM-BiG-AMP-2 estimated background and foreground frames in the middle and right columns, respectively. We note that, for this sequence, the rank-contraction strategy reduced the rank of the background component to 1 after the first EM iteration. Similar results (not shown here for reasons of space) were obtained with other video sequences.

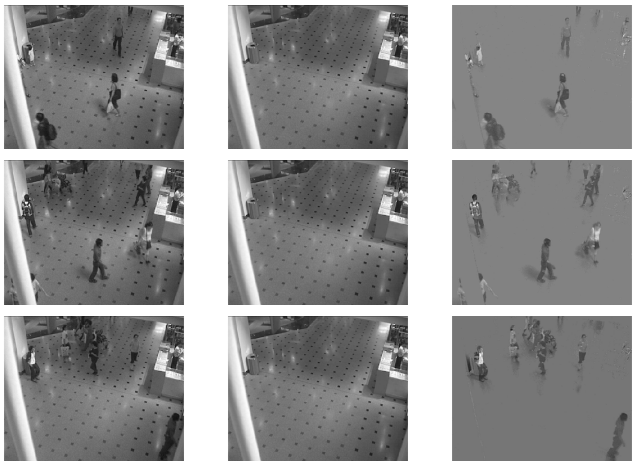

Figure 4.11: Three example frames from the "mall" video sequence. The left column shows original frames, the middle column EM-BiG-AMP-2 estimated background, and the right column EM-BiG-AMP-2 estimated foreground.

[37]The maximum allowed damping was reduced to `stepMax` $= 0.125$ for this experiment. To reduce runtime, a relatively loose tolerance of $5 \times 10^{-4}$ was used to establish EM and BiG-AMP convergence.

[38]See `http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html`.

## 4.8 Dictionary Learning

### 4.8.1 Problem setup

In dictionary learning (DL) [83], one seeks a dictionary $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ that allows the training samples $\boldsymbol{Y} \in \mathbb{R}^{M \times L}$ to be encoded as $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{W}$ for some sparse coefficient matrix $\boldsymbol{X}$ and small perturbation $\boldsymbol{W}$. One chooses $N = M$ to learn a square dictionary or $N > M$ (where $N$ is often a small multiple of $M$) to learn an overcomplete dictionary. In general, one must have a sufficient number of training examples, $L \gg N$, to avoid over-fitting.[39]

The BiG-AMP methodology is particularly well-suited to the DL problem, since both are inherently bilinear. In this work, for simplicity, we model the entries of $\mathsf{A}$ using the iid standard normal prior (4.117) and the entries of $\mathsf{X}$ using the iid zero-mean Bernoulli-Gaussian (BG) prior

$$p_{\mathsf{x}_{nl}}(x) = (1-\xi)\delta(x) + \xi\mathcal{N}(x; 0, \nu^x), \tag{4.136}$$

where $\xi$ represents the activity rate and $\nu^x$ the active-component variance. However, other priors could be considered, such as truncated Gaussian mixtures with column-dependent prior parameters in the case of non-negative matrix factorization [64]. For the likelihood $p_{\mathsf{Y}|\mathsf{Z}}$, we again select the PIAWGN model (4.119), but note that in most applications of DL the observations are completely observed.

### 4.8.2 Initialization

In general, we advocate initializing $\widehat{x}_{nl}(1)$ at the mean of the assumed prior on $\mathsf{x}_{nl}$, and initializing the variances $\nu^x_{nl}(1)$ and $\nu^a_{mn}(1)$ at 10 times the variance of $\mathsf{x}_{nl}$

[39] See [84] for a discussion of sample-size requirements for exact recovery of square dictionaries.

and $\mathsf{a}_{mn}$, respectively. We now discuss several strategies for initializing the dictionary estimates $\widehat{a}_{mn}(1)$. One option is to draw $\widehat{a}_{mn}(1)$ randomly from the assumed prior on $\mathsf{a}_{mn}$, as suggested for MC and RPCA. Although this approach works reasonably well, the prevalence of local minima in the DL problem motivates us to propose two alternative strategies. The first alternative is to exploit prior knowledge of a "good" sparsifying dictionary $\boldsymbol{A}$, in the case that such knowledge exists. With natural images, for example, the discrete cosine transform (DCT) and discrete wavelet transform (DWT) are known to yield reasonably sparse transform coefficients, and so the DCT and DWT matrices make appropriate initializations of $\widehat{\boldsymbol{A}}(1)$.

The second alternative is to initialize $\widehat{\boldsymbol{A}}(1)$ using an appropriately chosen subset of the columns of $\boldsymbol{Y}$, which is well motivated in the case that there exists a very sparse representation $\boldsymbol{X}$. For example, if there existed a decomposition $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}$ in which $\boldsymbol{X}$ had 1-sparse columns, then the columns of $\boldsymbol{A}$ would indeed match a subset of the columns of $\boldsymbol{Y}$ (up to a scale factor). In the general case, however, it is not apriori obvious which columns of $\boldsymbol{Y}$ to choose, and so we suggest the following greedy heuristic, which aims for a well-conditioned $\widehat{\boldsymbol{A}}(1)$: select (normalized) columns from $\boldsymbol{Y}$ sequentially, in random order, accepting each candidate if the mutual coherences with the previously selected columns and the condition number of the resulting submatrix are all sufficiently small. If all columns of $\boldsymbol{Y}$ are examined before finding $N$ acceptable candidates, then the process is repeated using a different random order. If repeated re-orderings fail, then $\widehat{\boldsymbol{A}}(1)$ is initialized using a random draw from $\mathbf{A}$.

### 4.8.3 EM-BiG-AMP

To tune the distributional parameters $\boldsymbol{\theta} = [\nu^w, \nu_0^x, \xi]^T$, we can straightforwardly apply the EM approach from Section 4.5.1. For this, we suggest initializing $\xi = 0.1$ (since Section 4.8.5 shows that this works well over a wide range of problems) and initializing $\nu_0^x$ and $\nu^w$ using a variation on the procedure suggested for MC that accounts for the sparsity of $\mathsf{x}_{nl}$:

$$\nu^w = \frac{\|P_\Omega(\boldsymbol{Y})\|_F^2}{(\mathrm{SNR}^0 + 1)|\Omega|} \tag{4.137}$$

$$\nu_0^x = \frac{1}{N\xi}\left[\frac{\|P_\Omega(\boldsymbol{Y})\|_F^2}{|\Omega|} - \nu^w\right]. \tag{4.138}$$

### 4.8.4 Avoiding Local Minima

The DL problem is fraught with local minima (see, e.g., [84]), and so it is common to run iterative algorithms several times from different random initializations. For BiG-AMP, we suggest keeping the result of one initialization over the previous if both[40] the residual error $\|\widehat{\boldsymbol{Z}} - \boldsymbol{Y}\|_F$ and the average sparsity (as measured by $\frac{1}{NL}\sum_{nl}\Pr\{\mathsf{x}_{nl} \neq 0 \,|\, \mathbf{Y} = \boldsymbol{Y}\}$) decrease.

### 4.8.5 Dictionary Learning Experiments

In this section, we numerically investigate the performance of EM-BiG-AMP for DL, as described in Section 4.8. Comparisons are made with the greedy K-SVD algorithm [85], the SPAMS implementation of the online approach [86], and the ER-SpUD(proj) approach for square dictionaries [84].

[40] As an alternative, if both the previous and current solutions achieve sufficiently small residual error, then only the average sparsity is considered in the comparison.

**Noiseless Square Dictionary Recovery**

We first investigate recovery of square (i.e., $N = M$) dictionaries from noise-free observations, repeating the experiment from [84]. For this, realizations of the true dictionary $\boldsymbol{A}$ were created by drawing elements independently from the standard normal distribution and subsequently scaling the columns to unit $\ell_2$ norm. Meanwhile, realizations of the true $\boldsymbol{X}$ were created by selecting $K$ entries in each column uniformly at random and drawing their values from the standard normal distribution, while setting all other entries to zero. Finally, the observations were constructed as $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X}$, from which the algorithms estimated $\boldsymbol{A}$ and $\boldsymbol{X}$ (up to a permutation and scale). The accuracy of the dictionary estimate $\widehat{\boldsymbol{A}}$ was quantified using the relative NMSE metric [84]

$$\text{NMSE}(\widehat{\boldsymbol{A}}) \triangleq \min_{\boldsymbol{J} \in \mathcal{J}} \frac{\|\widehat{\boldsymbol{A}}\boldsymbol{J} - \boldsymbol{A}\|_F^2}{\|\boldsymbol{A}\|_F^2}, \tag{4.139}$$

where $\boldsymbol{J}$ is a generalized permutation matrix used to resolve the permutation and scale ambiguities.

The subplots on the left of Fig. 4.12 show the mean NMSE achieved by K-SVD, SPAMS, ER-SpUD(proj), and EM-BiG-AMP,[41] respectively, over 50 problem realizations, for various combinations of dictionary size $N \in \{10, \dots, 60\}$ and data sparsity $K \in \{1, \dots, 10\}$, using $L = 5N \log N$ training examples. K-SVD, SPAMS, and EM-BiG-AMP were run with 10 different random initializations for each problem realization. To choose among these initializations, EM-BiG-AMP used the procedure

[41] EM-BiG-AMP was allowed up to 20 EM iterations, with each EM iteration allowed a minimum of 30 and a maximum of 1500 BiG-AMP iterations. K-SVD was allowed up to 100 iterations and provided with knowledge of the true sparsity $K$. SPAMS was allowed 1000 iterations and run using the hand-tuned penalty $\lambda = 0.1/\sqrt{N}$. The non-iterative ER-SpUD(proj) was run using code provided by the authors without modification.

from Section 4.8.4, while K-SVD and SPAMS used oracle knowledge to choose the NMSE-minimizing initialization.

The left column in Fig. 4.12 shows that the K-SVD, ER-SpUD(proj), and EM-BiG-AMP algorithms all exhibit a relatively sharp phase-transition curve (PTC) separating success and failure regions, and that ER-SpUD(proj)'s PTC is the best, while EM-BiG-AMP's PTC is very similar. Meanwhile, K-SVD's PTC is much worse and SPAMS performance is not good enough to yield a sharp phase transition,[42] despite the fact that both use oracle knowledge. EM-BiG-AMP, by contrast, was not provided with any knowledge of the DL problem parameters, such as the true sparsity or noise variance (in this case, zero).

For the same problem realizations, Fig. 4.13 shows the runtime to NMSE $=$ $-60$ dB (measured using MATLAB's `tic` and `toc`) versus dictionary size $N$. The results show that EM-BG-AMP runs within an order-of-magnitude of the fastest algorithm (SPAMS) and more than two orders-of-magnitude faster than ER-SpUD(proj)[43] for larger dictionaries.

**Noisy Square Dictionary Recovery**

Next we examined the recovery of square dictionaries from AWGN-corrupted observations. For this, we repeated the experiment from the previous section, but constructed the observations as $\boldsymbol{Y} = \boldsymbol{Z} + \boldsymbol{W}$, where $\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{X}$ and $\boldsymbol{W}$ contained AWGN samples with variance adjusted to achieve an SNR $= \mathrm{E}\{\sum_{m,l} |\mathsf{z}_{ml}|^2\} / \mathrm{E}\{\sum_{m,l} |\mathsf{y}_{ml} - \mathsf{z}_{ml}|^2\}$ of 40 dB.

[42]Our results for SPAMS and ER-SPUD(proj) in the left column of Fig. 4.12 are nearly identical to those in [84, Fig. 1], while our results for K-SVD are noticeably better.

[43]The simpler "SC" variant of ER-SpUD reduces the computational cost relative to the "proj" variant, but results in a significantly worse PTC (see [84, Fig. 1]) and remains slower than EM-BiG-AMP for larger problems.

The right subplots in Fig. 4.12 show the mean value (over 10 trials) of the relative NMSE from (4.139) when recovering an $N \times N$ dictionary from $L = 5N \log N$ training samples of sparsity $K$, for various combinations of $N$ and $K$. These subplots show that ER-SpUD(proj) falls apart in the noisy case, which is perhaps not surprising given that it is intended only for noiseless problems. Meanwhile, the K-SVD, SPAMS, and EM-BiG-AMP algorithms appear to degrade gracefully in the presence of noise, yielding NMSE $\approx -50$ dB at points below the noiseless PTCs.

**Recovery of Overcomplete Dictionaries**

Finally, we consider recovery of overcomplete $M \times N$ dictionaries, i.e., the case where $M < N$. In particular, we investigated the twice overcomplete case, $N = 2M$. For this, random problem realizations were constructed in the same manner as described earlier, except for the dictionary dimensions.

The left column of Fig. 4.14 shows the mean value (over 10 trials) of the relative NMSE for *noiseless* recovery, while the right column shows the corresponding results for *noisy* recovery. In all cases, $L = 5N \log N = 10M \log(2M)$ training samples were provided. EM-BiG-AMP, K-SVD, and SPAMS all give very similar results to Fig. 4.12 for the square-dictionary case, verifying that these techniques are equally suited to the recovery of over-complete dictionaries. ER-SpUD(proj), however, only applies to square dictionaries and hence was not tested here.

**Summary**

In summary, Figs. 4.12-4.14 show that, for noiseless square dictionary learning, EM-BiG-AMP yields an empirical PTC that is nearly as good as the state-of-the-art ER-SpUD(proj) algorithm and much better than those of (genie-aided) K-SVD

and SPAMS. However, the figures show that, in comparison to ER-SpUD(proj), EM-BiG-AMP is fast for large (square) dictionaries, robust to AWGN, and applicable to non-square dictionaries.

We recall that Krzakala, Mézard, and Zdeborová recently proposed an AMP-based approach to blind calibration and dictionary learning [68] that bears similarity to our scalar-variance BiG-AMP under AWGN-corrupted observations (recall footnote 16). Although their approach gave good results for blind calibration, they report that it was "not able to solve" the DL problem [68]. We attribute EM-BiG-AMP's success with DL (as evidenced by Figs. 4.12-4.14) to the adaptive damping procedure proposed in Section 4.4.2, the initialization procedure proposed in Section 4.8.2, the EM-learning procedure proposed in Section 4.8.3, and the re-initialization procedure proposed in Section 4.8.4.

## 4.9 Conclusion

In this chapter, we presented BiG-AMP, an extension of the GAMP algorithm proposed for high-dimensional generalized-*linear* regression in the context of compressive sensing, to generalized-*bilinear* regression, with applications in matrix completion, robust PCA, dictionary learning, and related matrix-factorization problems. In addition, we proposed an adaptive damping mechanism to aid convergence under realistic problem sizes, an expectation-maximization (EM)-based method to automatically tune the parameters of the assumed priors, and two rank-selection strategies. Extensive numerical results, conducted with synthetic and realistic datasets for matrix completion, robust PCA, and dictionary learning problems, demonstrated that

BiG-AMP yields excellent reconstruction accuracy (often best in class) while maintaining competitive runtimes, and that the proposed EM and rank-selection strategies successfully avoid the need to tune algorithmic parameters.

The excellent empirical results reported here motivate future work on the analysis of EM-BiG-AMP, on the extension of EM-BiG-AMP to, e.g., structured-sparse or parametric models, and on the application of EM-BiG-AMP to practical problems in high-dimensional inference. For example, preliminary results on the application of EM-BiG-AMP to hyperspectral unmixing have been reported in [64] and are very encouraging.

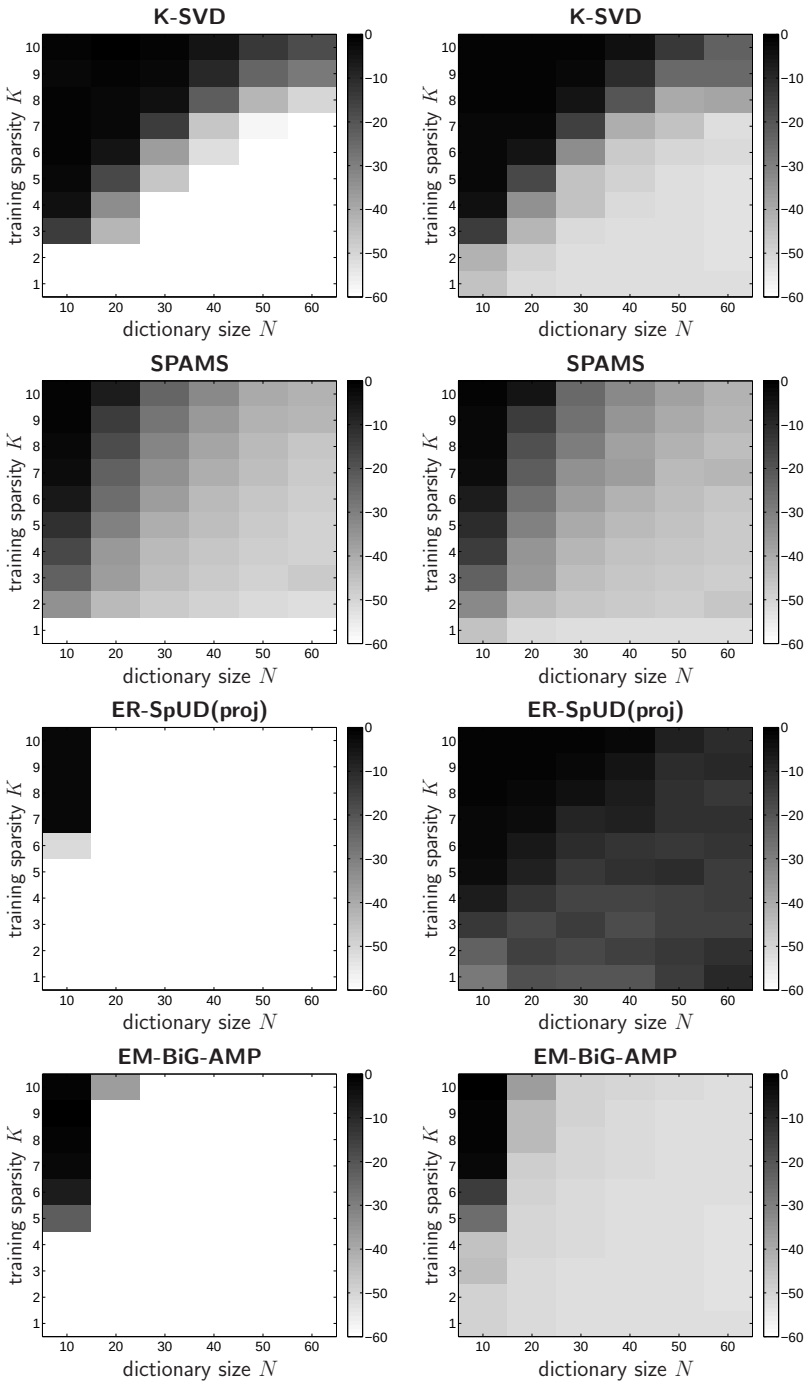


Figure 4.12: Mean NMSE (over 10 trials) for recovery of an $N \times N$ dictionary from $L = 5N \log N$ training samples, each of sparsity $K$, in the noiseless case (left) and under AWGN of 40 dB SNR (right), for several algorithms.

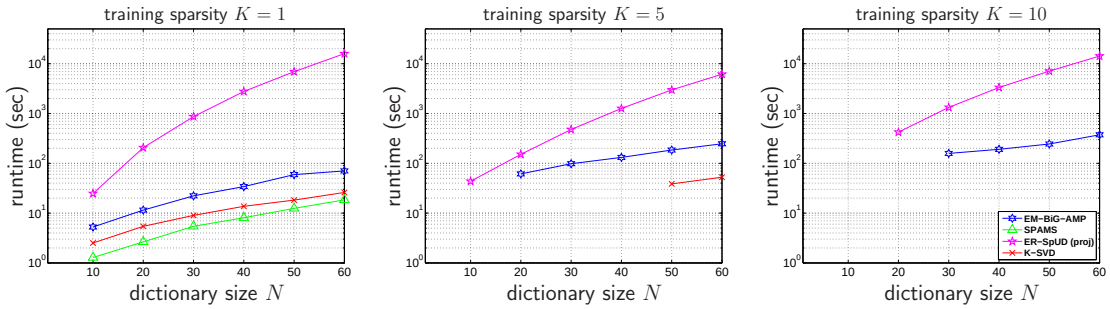


Figure 4.13: Median runtime until termination (over 10 trials) versus dictionary size $N$, for noiseless recovery of a square dictionary from $L = 5N \log N$ $K$-sparse samples, for several values of training sparsity $K$. Missing values indicate that the algorithm did not achieve the required NMSE before termination and correspond to the black regions in the panes on the left of Fig. 4.12.

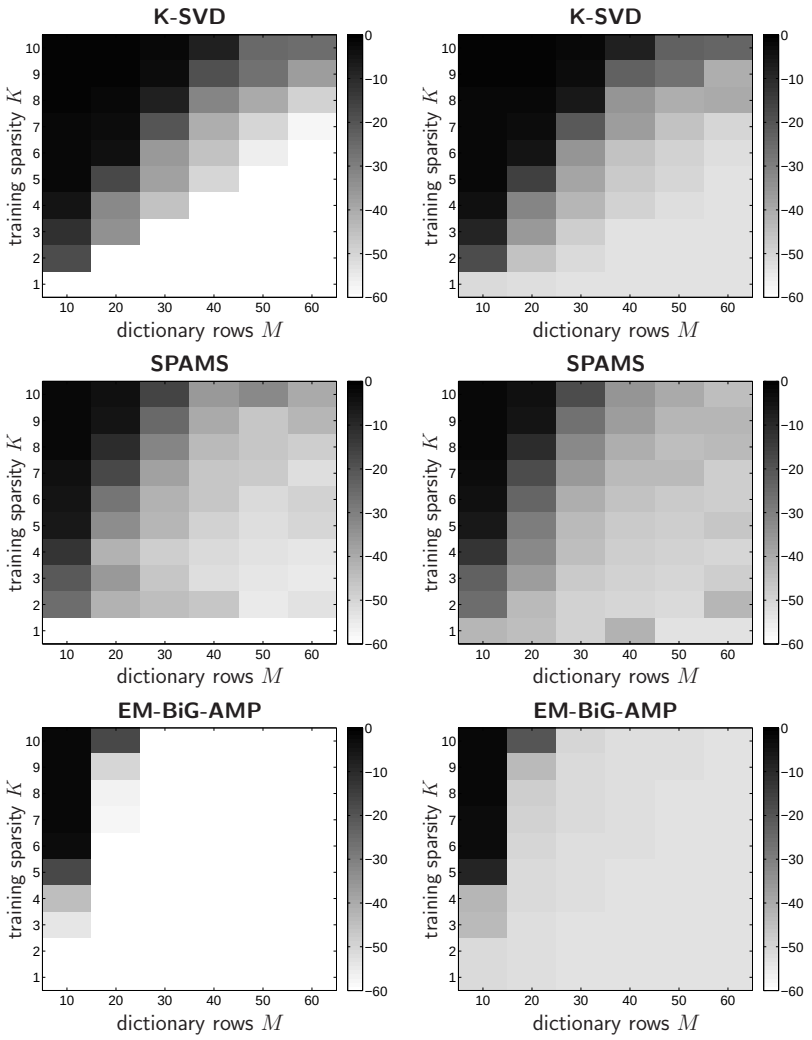


Figure 4.14: Mean NMSE (over 10 trials) for recovery of an $M \times (2M)$ dictionary from $L = 10M\log(2M)$ training samples, each of sparsity $K$, in the noiseless case (left) and under AWGN of 40 dB SNR (right), for several algorithms.

# Chapter 5: Parametric Bilinear Generalized Approximate Message Passing

## 5.1 Overview

In this chapter, we propose an approximate-message-passing [16] based algorithmic framework for a class of *parametric generalized bilinear* inference problems. In particular, we seek to estimate the parameter vectors $\boldsymbol{b} \in \mathbb{R}^{N_b}$ and $\boldsymbol{c} \in \mathbb{R}^{N_c}$ from a matrix observation $\boldsymbol{Y} \in \mathbb{R}^{M \times L}$ under a likelihood model of the form $p_{\mathbf{Y}|\mathbf{Z}}(\boldsymbol{Y} \mid \boldsymbol{A}(\boldsymbol{b})\boldsymbol{X}(\boldsymbol{c}))$, where $\boldsymbol{A}(\cdot) : \mathbb{R}^{N_b} \to \mathbb{R}^{M \times N}$ and $\boldsymbol{X}(\cdot) : \mathbb{R}^{N_c} \to \mathbb{R}^{N \times L}$ are known matrix-valued parameterizations. In doing so, we treat $\boldsymbol{b}$ and $\boldsymbol{c}$ as realizations of random vectors $\mathbf{b}$ and $\mathbf{c}$ with independent components, i.e.,

$$p_{\mathbf{b},\mathbf{c}}(\boldsymbol{b}, \boldsymbol{c}) = \prod_{i=1}^{N_b} p_{\mathsf{b}_i}(b_i) \prod_{j=1}^{N_c} p_{\mathsf{c}_j}(c_j), \tag{5.1}$$

and we assume that the parameterized bilinear form

$$\boldsymbol{Z} \triangleq \boldsymbol{A}(\boldsymbol{b})\boldsymbol{X}(\boldsymbol{c}) \tag{5.2}$$

has a known and separable likelihood function

$$p_{\mathbf{Y}|\mathbf{Z}}(\boldsymbol{Y} \mid \boldsymbol{Z}) = \prod_{m=1}^{M} \prod_{l=1}^{L} p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml} \mid z_{ml}) \tag{5.3}$$

where $\mathsf{Z} \triangleq \boldsymbol{A}(\mathsf{b})\boldsymbol{X}(\mathsf{c})$ is the random representation of $\boldsymbol{Z}$.

The above model can be recognized as a parametric extension of the *generalized bilinear* inference problem in [8, 9]. There, the goal was to infer the matrices $\boldsymbol{A}$ and $\boldsymbol{X}$ from observations $\boldsymbol{Y}$ under a likelihood model $p_{\mathsf{Y}|\mathsf{Z}}(\boldsymbol{Y} \,|\, \boldsymbol{A}\boldsymbol{X})$ with separable $p_{\mathsf{Y}|\mathsf{Z}}(\cdot|\cdot)$, $p_{\mathsf{A}}(\cdot)$ and $p_{\mathsf{X}}(\cdot)$. In this work, the matrices $\boldsymbol{A}$ and $\boldsymbol{X}$ are parameterized by random vectors $\mathsf{b}$ and $\mathsf{c}$ with separable priors, which allows the priors on $\boldsymbol{A}(\mathsf{b})$ and $\boldsymbol{X}(\mathsf{c})$ to be non-separable. In the interest of generality, we carry out the derivation for possibly non-linear parameterizations $\boldsymbol{A}(\cdot)$ and $\boldsymbol{X}(\cdot)$. However, certain steps in the derivation are rigorously justified only in the case of random linear $\boldsymbol{A}(\cdot)$ and $\boldsymbol{X}(\cdot)$, as will be detailed in the sequel. Furthermore, as we show in Section 5.6.2, the algorithm derived here reduces to the BiG-AMP algorithm from [8] for "trivial" deterministic parameterizations, i.e., when the elements in $\boldsymbol{A}(\boldsymbol{b})$ can be put in one-to-one correspondence with $\boldsymbol{b}$ and those in $\boldsymbol{X}(\boldsymbol{c})$ can be put in one-to-one correspondence with $\boldsymbol{c}$.

*Notation*: Throughout, we use san-serif font (e.g., $\mathsf{x}$) for random variables and serif font (e.g., $x$) otherwise. We use boldface capital letters (e.g., $\mathbf{X}$ and $\boldsymbol{X}$) for matrices, boldface small letters (e.g., $\mathbf{x}$ and $\boldsymbol{x}$) for vectors, and non-bold small letters (e.g., $\mathsf{x}$ and $x$) for scalars. We then use $p_{\mathsf{x}}(x)$ to denote the pdf of random quantity $\mathsf{x}$, and $\mathcal{N}(x; \widehat{x}, \nu^x)$ to denote the Gaussian pdf for a scalar random variable with mean $\widehat{x}$ and variance $\nu^x$. Also, we use $\mathrm{E}\{\mathsf{x}\}$ and $\mathrm{var}\{\mathsf{x}\}$ to denote mean and variance of $\mathsf{x}$, respectively, and $D(p_1 \| p_2)$ for the Kullback-Leibler (KL) divergence between pdfs $p_1$ and $p_2$. For matrices $\boldsymbol{X}$, we use $x_{nl} = [\boldsymbol{X}]_{nl}$ to denote the entry in the $n^{th}$ row and $l^{th}$ column, $\|\boldsymbol{X}\|_F$ to denote the Frobenius norm, and $\boldsymbol{X}^{\mathsf{T}}$ to denote the transpose.

For vectors $\boldsymbol{x}$, we use $x_n = [\boldsymbol{x}]_n$ to denote the $n^{th}$ entry and $\|\boldsymbol{x}\|_p = (\sum_n |x_n|^p)^{1/p}$ to denote the $\ell_p$ norm.

## 5.2 The Parameterizations

### 5.2.1 Random Affine Parameterizations

We are motivated by applications where $\boldsymbol{A}(\cdot)$ and $\boldsymbol{X}(\cdot)$ are smooth parameterizations, allowing first-order Taylor series approximations about $\widehat{\boldsymbol{b}} \in \mathbb{R}^{N_b}$ and $\widehat{\boldsymbol{c}} \in \mathbb{R}^{N_c}$ that take the form

$$a_{mn}(\boldsymbol{b}) \approx a_{mn}(\widehat{\boldsymbol{b}}) + \sum_{i=1}^{N_b} (b_i - \widehat{b}_i) a_{mn}^{(i)}(\widehat{\boldsymbol{b}}) \tag{5.4}$$

$$x_{nl}(\boldsymbol{c}) \approx x_{nl}(\widehat{\boldsymbol{c}}) + \sum_{j=1}^{N_c} (c_j - \widehat{c}_j) x_{nl}^{(j)}(\widehat{\boldsymbol{c}}), \tag{5.5}$$

where $a_{mn}^{(i)} \triangleq \frac{\partial}{\partial b_i} a_{mn}(\boldsymbol{b})$ and $x_{nl}^{(j)} \triangleq \frac{\partial}{\partial c_j} x_{nl}(\boldsymbol{c})$. However, to facilitate our AMP-based derivation, we assume that $\boldsymbol{A}(\cdot)$ and $\boldsymbol{X}(\cdot)$ are *random affine parameterizations* of the form

$$a_{mn}(\boldsymbol{b}) = \frac{1}{\sqrt{N_b}} a_{mn}^{(0)} + \sum_{i=1}^{N_b} b_i a_{mn}^{(i)} = \sum_{i=0}^{N_b} b_i a_{mn}^{(i)} \tag{5.6}$$

$$x_{nl}(\boldsymbol{c}) = \frac{1}{\sqrt{N_c}} x_{nl}^{(0)} + \sum_{j=1}^{N_c} c_j x_{nl}^{(j)} = \sum_{j=0}^{N_c} c_j x_{nl}^{(j)}, \tag{5.7}$$

where $a_{mn}^{(i)}$ and $x_{nl}^{(j)}$ are realizations of independent zero-mean random variables $\mathsf{a}_{mn}^{(i)}$ and $\mathsf{x}_{nl}^{(j)}$, respectively, with $\mathrm{E}\{(\mathsf{a}_{mn}^{(i)})^2\} = O(1/N)$ and $\mathrm{E}\{(\mathsf{x}_{nl}^{(j)})^2\} = O(1)$. For the second equality in (5.6)-(5.7), we used $b_0 \triangleq 1/\sqrt{N_b}$ and $c_0 \triangleq 1/\sqrt{N_c}$. Note that, for any $\widehat{\boldsymbol{b}}$ and $\widehat{\boldsymbol{c}}$, (5.6)-(5.7) can be rewritten as

$$a_{mn}(\boldsymbol{b}) = a_{mn}(\widehat{\boldsymbol{b}}) + \sum_{i=1}^{N_b} (b_i - \widehat{b}_i) a_{mn}^{(i)} \tag{5.8}$$

$$x_{nl}(\boldsymbol{c}) = x_{nl}(\widehat{\boldsymbol{c}}) + \sum_{j=1}^{N_c} (c_j - \widehat{c}_j) x_{nl}^{(j)}, \tag{5.9}$$

which are reminiscent of (5.4)-(5.5) except that (5.8)-(5.9) use equalities and location-invariant partial derivatives.

### 5.2.2 Large-System Limit Scalings

Our AMP-based derivation will rely primarily on central-limit-theorem (CLT) and Taylor-series approximations that become exact in the large-system limit (LSL), where $N, M, L, N_b, N_c \to \infty$ such that $M/N$, $L/N$, $N_b/N^2$, and $N_c/N^2$ converge to fixed positive constants.

In the sequel, we will assume that both $\mathrm{E}\{\mathsf{b}_i^2\}$ and $\mathrm{E}\{\mathsf{c}_j^2\}$ scale as $O(1/N^2)$. Then, invoking the independence among the elements in $\mathbf{b}$ and $\mathbf{c}$ implied by (5.1), the random affine model (5.6)-(5.7) yields

$$\mathrm{E}\{\mathsf{a}_{mn}(\mathbf{b})^2\} = \mathrm{E}\left\{ \sum_{i=0}^{N_b} \mathsf{b}_i \mathsf{a}_{mn}^{(i)} \sum_{k=0}^{N_b} \mathsf{b}_k \mathsf{a}_{mn}^{(k)} \right\} \tag{5.10}$$

$$= \sum_{i=0}^{N_b} \left[ \mathrm{E}\{(\mathsf{a}_{mn}^{(i)})^2\}\, \mathrm{E}\{\mathsf{b}_i^2\} + \sum_{k\neq i} \underbrace{\mathrm{E}\{\mathsf{a}_{mn}^{(i)}\}\, \mathrm{E}\{\mathsf{a}_{mn}^{(k)}\}}_{=0}\, \mathrm{E}\{\mathsf{b}_i\}\, \mathrm{E}\{\mathsf{b}_k\} \right] \tag{5.11}$$

$$= O(1/N) \tag{5.12}$$

$$\mathrm{E}\{\mathsf{x}_{nl}(\mathbf{c})^2\} = \mathrm{E}\left\{ \sum_{j=0}^{N_c} \mathsf{c}_j \mathsf{x}_{nl}^{(j)} \sum_{k=0}^{N_c} \mathsf{c}_k \mathsf{x}_{nl}^{(k)} \right\} \tag{5.13}$$

$$= \sum_{j=0}^{N_c} \left[ \mathrm{E}\{(\mathsf{x}_{nl}^{(j)})^2\}\, \mathrm{E}\{\mathsf{c}_j^2\} + \sum_{k\neq j} \underbrace{\mathrm{E}\{\mathsf{x}_{nl}^{(j)}\}\, \mathrm{E}\{\mathsf{x}_{nl}^{(k)}\}}_{=0}\, \mathrm{E}\{\mathsf{c}_j\}\, \mathrm{E}\{\mathsf{c}_k\} \right] \tag{5.14}$$

$$= O(1), \tag{5.15}$$

which conveniently matches the scaling assumptions on $\mathbf{A}$ and $\mathbf{X}$ employed in the BiG-AMP derivation [8]. Similarly, for the bilinear form $\mathsf{z}_{ml} = \sum_{n=1}^{N} \mathsf{a}_{mn}(\mathbf{b})\mathsf{x}_{nl}(\mathbf{c})$,

we have

$$\mathrm{E}\{\mathsf{z}_{ml}^2\} = \mathrm{E}\left\{ \sum_{n=1}^{N} \sum_{i=0}^{N_b} \mathsf{b}_i \mathsf{a}_{mn}^{(i)} \sum_{j=0}^{N_c} \mathsf{c}_j \mathsf{x}_{nl}^{(j)} \sum_{k=1}^{N} \sum_{r=0}^{N_b} \mathsf{b}_r \mathsf{a}_{mk}^{(r)} \sum_{q=0}^{N_c} \mathsf{c}_q \mathsf{x}_{kl}^{(q)} \right\} \tag{5.16}$$

$$= \sum_{n=1}^{N} \Bigg[ \sum_{i,r} \mathrm{E}\{\mathsf{b}_i \mathsf{b}_r\} \, \mathrm{E}\{\mathsf{a}_{mn}^{(i)} \mathsf{a}_{mn}^{(r)}\} \sum_{j,q} \mathrm{E}\{\mathsf{c}_j \mathsf{c}_q\} \, \mathrm{E}\{\mathsf{x}_{nl}^{(j)} \mathsf{x}_{nl}^{(q)}\}$$

$$+ \sum_{k \neq n} \sum_{i,r,j,q} \mathrm{E}\{\mathsf{b}_i \mathsf{b}_r\} \underbrace{\mathrm{E}\{\mathsf{a}_{mn}^{(i)}\} \, \mathrm{E}\{\mathsf{a}_{mk}^{(r)}\}}_{=0} \mathrm{E}\{\mathsf{c}_j \mathsf{c}_q\} \underbrace{\mathrm{E}\{\mathsf{x}_{nl}^{(j)}\} \, \mathrm{E}\{\mathsf{x}_{kl}^{(q)}\}}_{=0} \Bigg] \tag{5.17}$$

$$= \sum_{n=1}^{N} \sum_{i=1}^{N_b} \left[ \mathrm{E}\{\mathsf{b}_i^2\} \, \mathrm{E}\{(\mathsf{a}_{mn}^{(i)})^2\} + \sum_{r \neq i} \mathrm{E}\{\mathsf{b}_i\} \, \mathrm{E}\{\mathsf{b}_r\} \underbrace{\mathrm{E}\{\mathsf{a}_{mn}^{(i)}\} \, \mathrm{E}\{\mathsf{a}_{mn}^{(r)}\}}_{=0} \right]$$

$$\times \sum_{j=1}^{N_c} \left[ \mathrm{E}\{\mathsf{c}_j^2\} \, \mathrm{E}\{(\mathsf{x}_{nl}^{(j)})^2\} + \sum_{q \neq j} \mathrm{E}\{\mathsf{c}_j\} \, \mathrm{E}\{\mathsf{c}_q\} \underbrace{\mathrm{E}\{\mathsf{x}_{nl}^{(j)}\} \, \mathrm{E}\{\mathsf{x}_{nl}^{(q)}\}}_{=0} \right] \tag{5.18}$$

$$= O(1), \tag{5.19}$$

which matches the scaling assumptions on $\mathsf{Z}$ used in BiG-AMP [8]. We note that various other choices of scalings on $\mathrm{E}\{(\mathsf{a}_{mn}^{(i)})^2\}$, $\mathrm{E}\{(\mathsf{x}_{nl}^{(j)})^2\}$, $\mathrm{E}\{\mathsf{b}_i^2\}$, and $\mathrm{E}\{\mathsf{c}_j^2\}$ yield the scalings on $\mathrm{E}\{\mathsf{a}_{mn}(\mathbf{b})^2\}$, $\mathrm{E}\{\mathsf{x}_{nl}(\mathbf{c})^2\}$, and $\mathrm{E}\{\mathsf{z}_{ml}^2\}$ in (5.12), (5.15), and (5.19). In thise sense, the choices that we made purely out of convenience.

## 5.3 Bayesian Inference

For the statistical model (5.1)-(5.3), the posterior pdf is

$$p_{\mathbf{b},\mathbf{c}|\mathbf{Y}}(\boldsymbol{b}, \boldsymbol{c} \,|\, \boldsymbol{Y}) = p_{\mathbf{Y}|\mathbf{b},\mathbf{c}}(\boldsymbol{Y} \,|\, \boldsymbol{b}, \boldsymbol{c}) \, p_{\mathbf{b}}(\boldsymbol{b}) \, p_{\mathbf{c}}(\boldsymbol{c}) / p_{\mathbf{Y}}(\boldsymbol{Y}) \tag{5.20}$$

$$\propto p_{\mathbf{Y}|\mathbf{Z}}(\boldsymbol{Y} \,|\, \boldsymbol{A}(\boldsymbol{b})\boldsymbol{X}(\boldsymbol{c})) \, p_{\mathbf{b}}(\boldsymbol{b}) \, p_{\mathbf{c}}(\boldsymbol{c}) \tag{5.21}$$

$$= \Big[ \prod_m \prod_l p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}\Big(y_{ml} \,\Big|\, \sum_k a_{mk}(\boldsymbol{b}) x_{kl}(\boldsymbol{c})\Big) \Big] \Big[ \prod_i p_{\mathsf{b}_i}(b_i) \Big] \Big[ \prod_j p_{\mathsf{c}_j}(c_j) \Big], \tag{5.22}$$

where (5.20) employs Bayes' rule and $\propto$ denotes equality up to a constant scale factor. The posterior distribution can be represented using a bipartite factor graph of the

form in Fig. 5.1. There, the factors of the pdf $p_{\mathsf{b},\mathsf{c}|\mathsf{Y}}$ in (5.22) are represented by "factor nodes" that appear as black boxes, and the random variables in (5.22) are represented by "variable nodes" that appear as white circles. Each variable node is connected to every factor node in which that variable appears. We note that the observed data $\{y_{ml}\}$ are treated as parameters of the $p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml}|\cdot)$ factor nodes, and not as random variables. Also, although Fig. 5.1 shows an edge between *every* $\mathsf{b}_i$ and $p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}$ node pair, a given edge will vanish when $a_{mn}(\cdot)$ does not depend on $\mathsf{b}_i$. A similar statement applies to the edge between $\mathsf{c}_j$ and $p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}$.

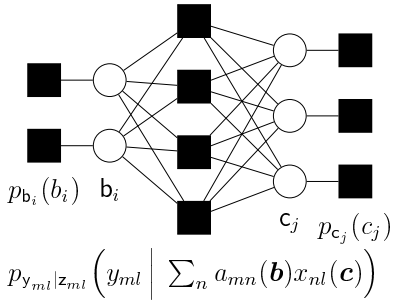


Figure 5.1: The factor graph for parametric generalized bilinear inference under $N_b = 2$, $N_c = 3$, and $ML = 4$.

## 5.4 Application of the Sum Product Algorithm

In our formulation of the SPA, four types of message will be used, as specified in Table 5.1. These messages take the form of log-pdfs with arbitrary constant offsets, which can be converted to pdfs via exponentiation and scaling. For example, message $\Delta^{\mathsf{b}}_{ml\to i}(t,.))$ corresponds to pdf $\frac{1}{C}\exp(\Delta^{\mathsf{b}}_{ml\to i}(t,.))$ with $C = \int_{b_i}\exp(\Delta^{\mathsf{b}}_{ml\to i}(t,b_i))$. Similarly, we express the (iteration-$t$ SPA-approximated) posterior pdfs $p_{\mathsf{b}_i|\mathsf{Y}}(t,.\,|\,\boldsymbol{Y})$

| | |
|---|---|
| $\Delta^{\mathsf{b}}_{ml\to i}(t,.)$ | SPA message from node $p_{\mathsf{y}_{ml}\vert\mathsf{z}_{ml}}$ to node $\mathsf{b}_i$ |
| $\Delta^{\mathsf{b}}_{ml\leftarrow i}(t,.)$ | SPA message from node $\mathsf{b}_i$ to node $p_{\mathsf{y}_{ml}\vert\mathsf{z}_{ml}}$ |
| $\Delta^{\mathsf{c}}_{ml\to j}(t,.)$ | SPA message from node $p_{\mathsf{y}_{ml}\vert\mathsf{z}_{ml}}$ to node $\mathsf{c}_j$ |
| $\Delta^{\mathsf{c}}_{ml\leftarrow j}(t,.)$ | SPA message from node $\mathsf{c}_j$ to node $p_{\mathsf{y}_{ml}\vert\mathsf{z}_{ml}}$ |
| $\Delta^{\mathsf{b}}_{i}(t,.)$ | SPA-approximated log posterior pdf of $\mathsf{b}_i$ |
| $\Delta^{\mathsf{c}}_{j}(t,.)$ | SPA-approximated log posterior pdf of $\mathsf{c}_j$ |

Table 5.1: SPA message definitions at iteration $t \in \mathbb{Z}$.

and $p_{\mathsf{c}_j\vert\mathbf{Y}}(t,.\,\vert\,\boldsymbol{Y})$ in the log domain as $\Delta^{\mathsf{b}}_i(t,.)$ and $\Delta^{\mathsf{c}}_j(t,.)$, respectively, again with arbitrary constant offsets.

Applying the SPA to the factor graph in Fig. 5.1, we arrive at the following update rules for the four messages in Table 5.1:

$$\Delta^{\mathsf{b}}_{ml\to i}(t,b_i) = \log \int_{\{b_r\}_{r\neq i},\{c_k\}_{k=1}^{N_c}} p_{\mathsf{y}_{ml}\vert\mathsf{z}_{ml}}\Big(y_{ml}\,\Big|\,\sum_k a_{mk}(\boldsymbol{b})x_{kl}(\boldsymbol{c})\Big) \times \prod_{r\neq i}\exp\Big(\Delta^{\mathsf{b}}_{ml\leftarrow r}(t,b_r)\Big)\prod_{k=1}^{N_c}\exp\Big(\Delta^{\mathsf{c}}_{ml\leftarrow k}(t,c_k)\Big) + \mathsf{const} \quad (5.23)$$

$$\Delta^{\mathsf{c}}_{ml\to j}(t,c_j) = \log \int_{\{b_r\}_{r=1}^{N_b},\{c_k\}_{k\neq j}} p_{\mathsf{y}_{ml}\vert\mathsf{z}_{ml}}\Big(y_{ml}\,\Big|\,\sum_k a_{mk}(\boldsymbol{b})x_{kl}(\boldsymbol{c})\Big) \times \prod_{r=1}^{N_b}\exp\Big(\Delta^{\mathsf{b}}_{ml\leftarrow r}(t,b_r)\Big)\prod_{k\neq j}\exp\Big(\Delta^{\mathsf{c}}_{ml\leftarrow k}(t,c_k)\Big) + \mathsf{const} \quad (5.24)$$

$$\Delta^{\mathsf{b}}_{ml\leftarrow i}(t+1,b_i) = \log p_{\mathsf{b}_i}(b_i) + \sum_{(r,k)\neq(m,l)}\Delta^{\mathsf{b}}_{rk\to i}(t,b_i) + \mathsf{const} \quad (5.25)$$

$$\Delta^{\mathsf{c}}_{ml\leftarrow j}(t+1,c_j) = \log p_{\mathsf{c}_j}(c_j) + \sum_{(r,k)\neq(m,l)}\Delta^{\mathsf{c}}_{rk\to j}(t,c_j) + \mathsf{const}, \quad (5.26)$$

where $\mathsf{const}$ denotes an arbitrary constant (w.r.t $b_i$ in (5.23) and (5.25) and w.r.t $c_j$ in (5.24) and (5.26)). In the sequel, we denote the mean and variance of the pdf $\frac{1}{C}\exp(\Delta^{\mathsf{b}}_{ml\leftarrow i}(t,.)$ by $\widehat{b}_{ml,i}(t)$ and $\nu^b_{ml,i}(t)$, respectively, and we denote the mean and variance of $\frac{1}{C}\exp(\Delta^{\mathsf{c}}_{ml\leftarrow j}(t,.))$ by $\widehat{c}_{ml,j}(t)$ and $\nu^c_{ml,j}(t)$. We refer to the vectors of these

statistics for a given $(m,l)$ as $\widehat{\boldsymbol{b}}_{ml}(t), \boldsymbol{\nu}^b_{ml}(t) \in \mathbb{R}^{N_b}$ and $\widehat{\boldsymbol{c}}_{ml}(t), \boldsymbol{\nu}^c_{ml}(t) \in \mathbb{R}^{N_c}$. For the log-posteriors, the SPA implies

$$\Delta^{\mathsf{b}}_i(t+1, b_i) = \log p_{\mathsf{b}_i}(b_i) + \sum_{m,l} \Delta^{\mathsf{b}}_{ml\to i}(t, b_i) + \mathsf{const} \tag{5.27}$$

$$\Delta^{\mathsf{c}}_j(t+1, c_j) = \log p_{\mathsf{c}_j}(c_j) + \sum_{m,l} \Delta^{\mathsf{c}}_{ml\to j}(t, c_j) + \mathsf{const}, \tag{5.28}$$

and we denote the mean and variance of $\frac{1}{C}\exp(\Delta^{\mathsf{b}}_i(t,.))$ by $\widehat{b}_i(t)$ and $\nu^b_i(t)$, and the mean and variance of $\frac{1}{C}\exp(\Delta^{\mathsf{c}}_j(t,.))$ by $\widehat{c}_j(t)$ and $\nu^c_j(t)$. Finally, we denote the vectors of these statistics as $\widehat{\boldsymbol{b}}(t), \boldsymbol{\nu}^b(t) \in \mathbb{R}^{N_b}$ and $\widehat{\boldsymbol{c}}(t), \boldsymbol{\nu}^c(t) \in \mathbb{R}^{N_c}$.

## 5.5 MMSE P-BiG-AMP Derivation

We now apply AMP approximations [6, 16] to the SPA updates (5.23)-(5.28). In particular, our derivation will neglect terms that vanish relative to others in the large-system limit, i.e., as $N \to \infty$ while simultaneously $M, L, N_b, N_c \to \infty$ such that $M/N$, $L/N$, $N_b/N^2$, and $N_c/N^2$ converge to fixed positive constants. In doing so, we assume the scalings on $\mathsf{b}_i$, $\mathsf{c}_j$, $a_{mn}(\mathbf{b})$, $x_{nl}(\mathbf{c})$, and $\mathsf{z}_{ml}$ that were established in Section 5.2.2, and we assume that the same scalings hold whether the random variables are distributed according to the priors, the SPA messages (5.23)-(5.26), or the SPA-approximated posteriors (5.27)-(5.28). These assumptions lead straightforwardly to the scalings of $\widehat{z}_{ml}(t)$, $\nu^z_{ml}(t)$, $\widehat{b}_{ml,i}(t)$, $\nu^b_{ml,i}(t)$, $\widehat{c}_{ml,j}(t)$, and $\nu^c_{ml,j}(t)$ specified in Table 5.2. Furthermore, we assume that both $\widehat{b}_{ml,i}(t) - \widehat{b}_i(t)$ and $\widehat{c}_{ml,j}(t) - \widehat{c}_j(t)$ are $O(1/N^2)$. Then similar reasoning leads to the assumed scalings on the variance differences in Table 5.2. These assumptions are similar to those used in previous AMP derivations [6, 16], particularly that of BiG-AMP [8]. Other entries in the table will be explained below.

| | | | | | |
|---|---|---|---|---|---|
| $a_{mn}(\mathbf{b})$ | $O(\frac{1}{\sqrt{N}})$ | $x_{nl}(\mathbf{c})$ | $O(1)$ | $\widehat{b}_{ml,i}(t) - \widehat{b}_i(t)$ | $O(\frac{1}{N^2})$ |
| $a_{mn}^{(i)}$ | $O(\frac{1}{\sqrt{N}})$ | $x_{nl}^{(j)}$ | $O(1)$ | $\widehat{c}_{ml,j}(t) - \widehat{c}_j(t)$ | $O(\frac{1}{N^2})$ |
| $\widehat{p}_{ml}(t)$ | $O(1)$ | $\nu^p_{ml}(t)$ | $O(1)$ | $\nu^b_{ml,i}(t) - \nu^b_i(t)$ | $O(\frac{1}{N^3})$ |
| $\widehat{b}_{ml,i}(t)$ | $O(\frac{1}{N})$ | $\nu^b_{ml,i}(t)$ | $O(\frac{1}{N^2})$ | $\nu^c_{ml,j}(t) - \nu^c_j(t)$ | $O(\frac{1}{N^3})$ |
| $\widehat{c}_{ml,j}(t)$ | $O(\frac{1}{N})$ | $\nu^c_{ml,j}(t)$ | $O(\frac{1}{N^2})$ | $x_{nl}(\widehat{\boldsymbol{c}}_{ml}(t)) - x_{nl}(\widehat{\boldsymbol{c}}(t))$ | $O(\frac{1}{N})$ |
| $\widehat{c}_{ml,j}(t)$ | $O(\frac{1}{N})$ | $\nu^c_{ml,j}(t)$ | $O(\frac{1}{N^2})$ | $a_{mn}(\widehat{\boldsymbol{b}}_{ml}(t)) - a_{mn}(\widehat{\boldsymbol{b}}(t))$ | $O(\frac{1}{N^{3/2}})$ |
| $\widehat{z}_{ml}(t)$ | $O(1)$ | $\nu^z_{ml}(t)$ | $O(1)$ | $\nu^r_{ml,j}(t) - \nu^r_j(t)$ | $O(\frac{1}{N^4})$ |
| $\widehat{z}^{(i,j)}_{\rightarrow ml}(t)$ | $O(1)$ | $\widehat{z}^{(*,j)}_{\rightarrow ml}(t)$ | $O(1)$ | $\widehat{r}_{ml,j}(t) - \widehat{r}_j(t)$ | $O(\frac{1}{N^2})$ |
| $\widehat{z}^{(i,*)}_{\rightarrow ml}(t)$ | $O(1)$ | $\widehat{z}^{(*,*)}_{ml}(t)$ | $O(1)$ | $\nu^q_{ml,i}(t) - \nu^q_i(t)$ | $O(\frac{1}{N^4})$ |
| $\widehat{r}_{ml,j}(t)$ | $O(\frac{1}{N})$ | $\nu^r_{ml,j}(t)$ | $O(\frac{1}{N^2})$ | $\widehat{q}_{ml,i}(t) - \widehat{q}_i(t)$ | $O(\frac{1}{N^2})$ |
| $\widehat{q}_{ml,i}(t)$ | $O(\frac{1}{N})$ | $\nu^q_{ml,i}(t)$ | $O(\frac{1}{N^2})$ | $\widehat{z}^{(*,j)}_{\rightarrow ml}(t) - \widehat{z}^{(*,j)}_{ml}(t)$ | $O(\frac{1}{N})$ |
| $\widehat{s}_{ml}(t)$ | $O(1)$ | $\nu^s_{ml}(t)$ | $O(1)$ | $\widehat{z}^{(i,*)}_{\rightarrow ml}(t) - \widehat{z}^{(i,*)}_{ml}(t)$ | $O(\frac{1}{N})$ |

Table 5.2: P-BiG-AMP variable scalings in the large-system limit.

### 5.5.1 SPA message from node $p_{\mathbf{y}_{ml}|\mathbf{z}_{ml}}$ to node $\mathbf{b}_i$

Consider the message defined in (5.23)

$$\Delta^{\mathsf{b}}_{ml\rightarrow i}(t, b_i) = \log \int_{\{b_r\}_{r\neq i}, \{c_k\}_{n=1}^{N_c}} p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}\Big(y_{ml} \,\Big|\, \overbrace{\sum_{n=1}^{N} a_{mn}(\boldsymbol{b}) x_{nl}(\boldsymbol{c})}^{z_{ml}}\Big) \times \prod_{r\neq i} \exp\Big(\Delta^{\mathsf{b}}_{ml\leftarrow r}(t, b_r)\Big) \prod_{k=1}^{N_c} \exp\Big(\Delta^{\mathsf{c}}_{ml\leftarrow k}(t, c_k)\Big) + \mathsf{const.} \tag{5.29}$$

The first step of our derivation invokes the large-system limit to apply the central limit theorem (CLT) to $\mathsf{z}_{ml}$, the random version of $z_{ml}$ in (5.29). In particular, the CLT motivates the treatment of $\mathsf{z}_{ml}$ conditioned on $\mathsf{b}_i = b_i$ as Gaussian and thus completely characterized by a (conditional) mean and variance, which we compute below.

Similar to previous AMP derivations, the use of the CLT can be justified in the case of the random affine parameterization described in Section 5.2.1. To see this, we write

$$\mathsf{z}_{ml} = \sum_{n=1}^{N}\sum_{i=0}^{N_b} \mathsf{b}_i a_{mn}^{(i)} \sum_{j=0}^{N_c} \mathsf{c}_j x_{nl}^{(j)} = \sum_{i=0}^{N_b}\mathsf{b}_i \sum_{j=0}^{N_c} \mathsf{c}_j \underbrace{\sum_{n=1}^{N} a_{mn}^{(i)} x_{nl}^{(j)}}_{\triangleq \widehat{z}_{ml}^{(i,j)}} = \mathbf{b}^{\mathsf{T}} \widehat{\boldsymbol{Z}}_{ml} \mathbf{c} \tag{5.30}$$

$$\begin{aligned} &= -\widehat{\boldsymbol{b}}_{ml}(t)^{\mathsf{T}} \widehat{\boldsymbol{Z}}_{ml} \widehat{\boldsymbol{c}}_{ml}(t) + \widehat{\boldsymbol{b}}_{ml}(t)^{\mathsf{T}} \widehat{\boldsymbol{Z}}_{ml} \mathbf{c} + \mathbf{b}^{\mathsf{T}} \widehat{\boldsymbol{Z}}_{ml} \widehat{\boldsymbol{c}}_{ml}(t) \\ &\quad + \big(\mathbf{b} - \widehat{\boldsymbol{b}}_{ml}(t)\big)^{\mathsf{T}} \widehat{\boldsymbol{Z}}_{ml} \big(\mathbf{c} - \widehat{\boldsymbol{c}}_{ml}(t)\big), \end{aligned} \tag{5.31}$$

where in (5.30) the matrix $\widehat{\boldsymbol{Z}}_{ml} \in \mathbb{R}^{N_b \times N_c}$ is constructed elementwise as $[\widehat{\boldsymbol{Z}}_{ml}]_{ij} = \widehat{z}_{ml}^{(i,j)}$, and in (5.31) we recall that $\widehat{\boldsymbol{b}}_{ml}(t)$ is the mean of $\mathbf{b}$ and $\widehat{\boldsymbol{c}}_{ml}(t)$ is the mean of $\mathbf{c}$ under the distribution in (5.29). Examining the terms in (5.31), we see that the first is an $O(1)$ constant, while the second and third are dense linear combinations of independent random variables that also scale as $O(1)$. As such, the second and third terms obey the central limit theorem, each converging in distribution to a Gaussian as $N \to \infty$. The last term in (5.31) can be written as a quadratic form in independent zero-mean random variables:

$$\big(\mathbf{b} - \widehat{\boldsymbol{b}}_{ml}(t)\big)^{\mathsf{T}} \widehat{\boldsymbol{Z}}_{ml} \big(\mathbf{c} - \widehat{\boldsymbol{c}}_{ml}(t)\big) = \begin{bmatrix} \mathbf{b} - \widehat{\boldsymbol{b}}_{ml}(t) \\ \mathbf{c} - \widehat{\boldsymbol{c}}_{ml}(t) \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} & \frac{1}{2}\widehat{\boldsymbol{Z}}_{ml} \\ \frac{1}{2}\boldsymbol{Z}_{ml}^{\mathsf{T}} & \end{bmatrix} \begin{bmatrix} \mathbf{b} - \widehat{\boldsymbol{b}}_{ml}(t) \\ \mathbf{c} - \widehat{\boldsymbol{c}}_{ml}(t) \end{bmatrix}. \tag{5.32}$$

It is shown in [87] that, for sufficiently dense $\boldsymbol{Z}_{ml}$ (as guaranteed by our random affine parameterization), the quadratic form in (5.31) converges in distribution to a zero-mean Gaussian as $N \to \infty$. Thus, in the large-system limit, $\mathsf{z}_{ml}$ equals a constant plus three Gaussian random variables, and thus $\mathsf{z}_{ml}$ is Gaussian.

Recalling the affine parameterization model assumed for our derivation and leveraging the convenient form (5.8)-(5.9), the conditional mean is

$$\mathrm{E}\{\mathsf{z}_{ml} \,|\, \mathsf{b}_i = b_i\} = \sum_{n=1}^{N} \mathrm{E}\{a_{mn}(\mathbf{b}) x_{nl}(\mathbf{c}) \,|\, \mathsf{b}_i = b_i\} \tag{5.33}$$

$$= \sum_{n=1}^{N} \mathrm{E}\left\{ \left[ a_{mn}\big(\widehat{\boldsymbol{b}}_{ml}(t)\big) + \sum_{i=1}^{N_b} \big(\mathsf{b}_i - \widehat{b}_{ml,i}(t)\big) a_{mn}^{(i)} \right] \right.$$
$$\left. \times \left[ x_{nl}\big(\widehat{\boldsymbol{c}}_{ml}(t)\big) + \sum_{j=1}^{N_c} \big(\mathsf{c}_j - \widehat{c}_{ml,j}(t)\big) x_{nl}^{(j)} \right] \,\middle|\, \mathsf{b}_i = b_i \right\} \tag{5.34}$$

$$= \sum_{n=1}^{N} \left[ a_{mn}\big(\widehat{\boldsymbol{b}}_{ml}(t)\big) + \big(b_i - \widehat{b}_{ml,i}(t)\big) a_{mn}^{(i)} \right] x_{nl}\big(\widehat{\boldsymbol{c}}_{ml}(t)\big) \tag{5.35}$$

$$= \underbrace{\sum_{n=1}^{N} \left[ a_{mn}\big(\widehat{\boldsymbol{b}}_{ml}(t)\big) - \widehat{b}_{ml,i}(t) a_{mn}^{(i)} \right] x_{nl}\big(\widehat{\boldsymbol{c}}_{ml}(t)\big)}_{\triangleq\, \widehat{p}_{i,ml}(t)} + b_i \sum_{n=1}^{N} a_{mn}^{(i)} x_{nl}\big(\widehat{\boldsymbol{c}}_{ml}(t)\big). \tag{5.36}$$

We note that the above definition of $\widehat{p}_{i,ml}(t)$ is conceptually similar to that used in BiG-AMP [8].

Likewise, the conditional variance is

$$
\begin{aligned}
&\operatorname{var}\{\mathsf{z}_{ml} \mid \mathsf{b}_i = b_i\} \\
&= \mathrm{E}\left\{ \left| \sum_{n=1}^{N} a_{mn}(\mathbf{b}) x_{nl}(\mathbf{c}) - \sum_{n=1}^{N} \mathrm{E}\{a_{mn}(\mathbf{b}) x_{nl}(\mathbf{c}) \mid \mathsf{b}_i = b_i\} \right|^2 \;\middle|\; \mathsf{b}_i = b_i \right\} && (5.37) \\
&= \mathrm{E}\left\{ \left| \sum_{n=1}^{N} \left( a_{mn}\big(\widehat{\boldsymbol{b}}_{ml}(t)\big) + \sum_{k=1}^{N_b} \big(\mathsf{b}_k - \widehat{b}_{ml,k}(t)\big) a_{mn}^{(k)} \right) \right. \right. \\
&\qquad \times \left( x_{nl}\big(\widehat{\boldsymbol{c}}_{ml}(t)\big) + \sum_{j=1}^{N_c} \big(\mathsf{c}_j - \widehat{c}_{ml,j}(t)\big) x_{nl}^{(j)} \right) \\
&\qquad \left. \left. - \left( \big(b_i - \widehat{b}_{ml,i}(t)\big) \sum_{n=1}^{N} a_{mn}^{(i)} x_{nl}\big(\widehat{\boldsymbol{c}}_{ml}(t)\big) + \sum_{n=1}^{N} a_{mn}\big(\widehat{\boldsymbol{b}}_{ml}(t)\big) x_{nl}\big(\widehat{\boldsymbol{c}}_{ml}(t)\big) \right) \right|^2 \;\middle|\; \mathsf{b}_i = b_i \right\} && (5.38) \\
&= \mathrm{E}\left\{ \left| \sum_{n=1}^{N} a_{mn}\big(\widehat{\boldsymbol{b}}_{ml}(t)\big) \sum_{j=1}^{N_c} \big(\mathsf{c}_j - \widehat{c}_{ml,j}(t)\big) x_{nl}^{(j)} + \sum_{n=1}^{N} \sum_{k=1}^{N_b} \big(\mathsf{b}_k - \widehat{b}_{ml,k}(t)\big) a_{mn}^{(k)} x_{nl}\big(\widehat{\boldsymbol{c}}_{ml}(t)\big) \right. \right. \\
&\qquad + \sum_{n=1}^{N} \sum_{k=1}^{N_b} \big(\mathsf{b}_k - \widehat{b}_{ml,k}(t)\big) a_{mn}^{(k)} \sum_{j=1}^{N_c} \big(\mathsf{c}_j - \widehat{c}_{ml,j}(t)\big) x_{nl}^{(j)} \\
&\qquad \left. \left. - \big(b_i - \widehat{b}_{ml,i}(t)\big) \sum_{n=1}^{N} a_{mn}^{(i)} x_{nl}\big(\widehat{\boldsymbol{c}}_{ml}(t)\big) \right|^2 \;\middle|\; \mathsf{b}_i = b_i \right\} && (5.39) \\
&= \mathrm{E}\left\{ \left| \sum_{j=1}^{N_c} \big(\mathsf{c}_j - \widehat{c}_{ml,j}(t)\big) \sum_{n=1}^{N} a_{mn}\big(\widehat{\boldsymbol{b}}_{ml}(t)\big) x_{nl}^{(j)} + \sum_{k \neq i} \big(\mathsf{b}_k - \widehat{b}_{ml,k}(t)\big) \sum_{n=1}^{N} a_{mn}^{(k)} x_{nl}\Big(\widehat{\boldsymbol{c}}_{ml}(t)\Big) \right. \right. \\
&\qquad \left. \left. + \left[ \big(b_i - \widehat{b}_{ml,i}(t)\big) + \sum_{k \neq i} \big(\mathsf{b}_k - \widehat{b}_{ml,k}(t)\big) \right] \sum_{j=1}^{N_c} \big(\mathsf{c}_j - \widehat{c}_{ml,j}(t)\big) \sum_{n=1}^{N} a_{mn}^{(k)} x_{nl}^{(j)} \right|^2 \right\} && (5.40) \\
&= \mathrm{E}\left\{ \left| \sum_{j=1}^{N_c} \big(\mathsf{c}_j - \widehat{c}_{ml,j}(t)\big) \widehat{z}_{\to ml}^{(*,j)}(t) + \sum_{k \neq i} \big(\mathsf{b}_k - \widehat{b}_{ml,k}(t)\big) \widehat{z}_{\to ml}^{(k,*)}(t) \right. \right. \\
&\qquad \left. \left. + \left[ \big(b_i - \widehat{b}_{ml,i}(t)\big) + \sum_{k \neq i} \big(\mathsf{b}_k - \widehat{b}_{ml,k}(t)\big) \right] \sum_{j=1}^{N_c} \big(\mathsf{c}_j - \widehat{c}_{ml,j}(t)\big) \widehat{z}_{ml}^{(i,j)} \right|^2 \right\}, && (5.41)
\end{aligned}
$$

where for (5.38) both (5.8)-(5.9) and (5.36) were used, for (5.39) sums were expanded and a term was cancelled, for (5.40) sums were rearranged and the conditioning on $\mathsf{b}_i = b_i$ was applied after which a term was cancelled, and for (5.41) the definitions in

(5.42)-(5.44) were applied, all of which are based on $z_{ml}(\boldsymbol{b},\boldsymbol{c}) \triangleq [\boldsymbol{A}(\boldsymbol{b})\boldsymbol{X}(\boldsymbol{c})]_{ml}$:

$$\widehat{z}^{(i,*)}_{\to ml}(t) \triangleq \frac{\partial}{\partial b_i} z_{ml}(\boldsymbol{b},\boldsymbol{c})\bigg|_{\boldsymbol{b}=\widehat{\boldsymbol{b}}_{ml}(t),\,\boldsymbol{c}=\widehat{\boldsymbol{c}}_{ml}(t)} = \sum_{n=1}^{N} a^{(i)}_{mn} x_{nl}\big(\widehat{\boldsymbol{c}}_{ml}(t)\big) \tag{5.42}$$

$$\widehat{z}^{(*,j)}_{\to ml}(t) \triangleq \frac{\partial}{\partial c_j} z_{ml}(\boldsymbol{b},\boldsymbol{c})\bigg|_{\boldsymbol{b}=\widehat{\boldsymbol{b}}_{ml}(t),\,\boldsymbol{c}=\widehat{\boldsymbol{c}}_{ml}(t)} = \sum_{n=1}^{N} a_{mn}\big(\widehat{\boldsymbol{b}}_{ml}(t)\big) x^{(j)}_{nl} \tag{5.43}$$

$$\widehat{z}^{(i,j)}_{ml} \triangleq \frac{\partial^2}{\partial b_i \partial c_j} z_{ml}(\boldsymbol{b},\boldsymbol{c})\bigg|_{\boldsymbol{b}=\widehat{\boldsymbol{b}}_{ml}(t),\,\boldsymbol{c}=\widehat{\boldsymbol{c}}_{ml}(t)} = \sum_{n=1}^{N} a^{(i)}_{mn} x^{(j)}_{nl}. \tag{5.44}$$

We note that (5.44) is independent of the iteration $t$ due to the location-invariant nature of the parameterization's partial derivatives under the affine model (5.8)-(5.9). Continuing from (5.41), we rearrange terms to obtain

$$\begin{aligned}
&\mathrm{var}\{\mathsf{z}_{ml} \,|\, \mathsf{b}_i = b_i\} \\
&= \mathrm{E}\left\{ \left| \sum_{j=1}^{N_c} \big(\mathsf{c}_j - \widehat{c}_{ml,j}(t)\big)\Big[\widehat{z}^{(*,j)}_{\to ml}(t) + \big(b_i - \widehat{b}_{ml,i}(t)\big)\widehat{z}^{(i,j)}_{ml}\Big] + \sum_{k\neq i} \big(\mathsf{b}_k - \widehat{b}_{ml,k}(t)\big)\widehat{z}^{(k,*)}_{\to ml}(t) \right.\right. \\
&\qquad \left.\left. + \sum_{k\neq i} \big(\mathsf{b}_k - \widehat{b}_{ml,k}(t)\big) \sum_{j=1}^{N_c} \big(\mathsf{c}_j - \widehat{c}_{ml,j}(t)\big)\widehat{z}^{(i,j)}_{ml} \right|^2 \right\} \qquad (5.45) \\
&= \sum_{j=1}^{N_c} \nu^{c}_{ml,j}(t)\Big[\widehat{z}^{(*,j)}_{\to ml}(t) + \big(b_i - \widehat{b}_{ml,i}(t)\big)\widehat{z}^{(i,j)}_{ml}\Big]^2 + \sum_{k\neq i} \nu^{b}_{ml,k}(t)\widehat{z}^{(k,*)}_{\to ml}(t)^2 \\
&\qquad + \sum_{k\neq i} \nu^{b}_{ml,k}(t) \sum_{j=1}^{N_c} \nu^{c}_{ml,j}(t)\widehat{z}^{(i,j)2}_{ml} \qquad (5.46) \\
&= \sum_{j=1}^{N_c} \nu^{c}_{ml,j}(t)\Big[\widehat{z}^{(*,j)}_{\to ml}(t)^2 + \widehat{b}_{ml,i}(t)^2\widehat{z}^{(i,j)2}_{ml} - 2\widehat{b}_{ml,i}(t)\widehat{z}^{(*,j)}_{\to ml}(t)\widehat{z}^{(i,j)}_{ml}\Big] \\
&\qquad + \sum_{k\neq i} \nu^{b}_{ml,k}(t)\left[\widehat{z}^{(k,*)}_{\to ml}(t)^2 + \sum_{j=1}^{N_c} \nu^{c}_{ml,j}(t)\widehat{z}^{(k,j)2}_{ml}\right] \\
&\qquad + b_i^2 \sum_{j=1}^{N_c} \nu^{c}_{ml,j}(t)\widehat{z}^{(i,j)2}_{ml} + 2b_i \sum_{j=1}^{N_c} \nu^{c}_{ml,j}(t)\left[\widehat{z}^{(*,j)}_{\to ml}(t)\widehat{z}^{(i,j)}_{ml} - \widehat{b}_{ml,i}(t)\widehat{z}^{(i,j)2}_{ml}\right] \qquad (5.47) \\
&= \nu^{p}_{i,ml}(t) + b_i^2 \sum_{j=1}^{N_c} \nu^{c}_{ml,j}(t)\widehat{z}^{(i,j)2}_{ml} + 2b_i \sum_{j=1}^{N_c} \nu^{c}_{ml,j}(t)\left[\widehat{z}^{(*,j)}_{\to ml}(t)\widehat{z}^{(i,j)}_{ml} - \widehat{b}_{ml,i}(t)\widehat{z}^{(i,j)2}_{ml}\right], \qquad (5.48)
\end{aligned}$$

where for (5.46) we exploited the zero-mean nature of $\mathsf{b}_i - \widehat{b}_{ml,i}(t)$ and $\mathsf{c}_j - \widehat{c}_{ml,j}(t)$ as well as the independence among the elements of $\left[\begin{smallmatrix}\mathsf{b}\\ \mathsf{c}\end{smallmatrix}\right]$, for (5.47) we expanded and rearranged terms, and for (5.48) we used the definition

$$\nu^p_{i,ml}(t) \triangleq \sum_{j=1}^{N_c} \nu^c_{ml,j}(t)\left[\widehat{z}^{(*,j)}_{\to ml}(t)^2 + \widehat{b}_{ml,i}(t)^2\widehat{z}^{(i,j)2}_{ml} - 2\widehat{b}_{ml,i}(t)\widehat{z}^{(*,j)}_{\to ml}(t)\widehat{z}^{(i,j)}_{ml}\right] + \sum_{k\neq i}\nu^b_{ml,k}(t)\left[\widehat{z}^{(k,*)}_{\to ml}(t)^2 + \sum_{j=1}^{N_c}\nu^c_{ml,j}(t)\widehat{z}^{(k,j)2}_{ml}\right]. \tag{5.49}$$

Using the scalings established in Table 5.2, and invoking independence among $\mathsf{b}_i$, $\mathsf{c}_j$, $\mathsf{a}^{(i)}_{mn}$, and $\mathsf{x}^{(j)}_{nl}$, it can be readily seen that $\widehat{p}_{i,ml}(t)$ in (5.36), the terms in (5.42)-(5.44), and $\nu^p_{i,ml}(t)$ all scale as $O(1)$. Since $\widehat{p}_{i,ml}(t)$ and $\nu^p_{i,ml}(t)$ correspond to the mean and variance of a distribution on $\mathsf{z}_{ml}$, this scaling is expected.

A Monte Carlo simulation was performed to double-check the derivation of the expressions in (5.36) and (5.48). For the simulation, $b_i$ was fixed at a particular value, after which $z_{ml} = \sum_{n=1}^{N}\sum_{k=0}^{N_b} b_k a^{(k)}_{mn}\sum_{j=0}^{N_c} c_j x^{(j)}_{nl}$ was computed using i.i.d zero-mean Gaussian realizations of $\mathsf{b}_{k\neq i}$, $\mathsf{c}_j$, $\mathsf{a}^{(i)}_{mn}$, and $\mathsf{x}^{(j)}_{nl}$. Using $10^4$ trials, the empirical mean and variance of the resulting $z_{ml}$ were computed and compared to the analytical expressions in (5.36) and (5.48). Figure 5.2 plots the these quantities for a range of $b_i$ values, showing a close agreement between empirical and analytical results. Under correct (5.36) and (5.48), this is expected: the Law of Large Numbers guarantees that the empirical and analytical values will converge as the number of Monte Carlo trials grows to infinity.

We now use the Gaussian approximation of $\mathsf{z}_{ml}|_{\mathsf{b}_i=b_i}$ (whose mean and variance appear in (5.36) and (5.48), respectively) to reduce the representation of the SPA

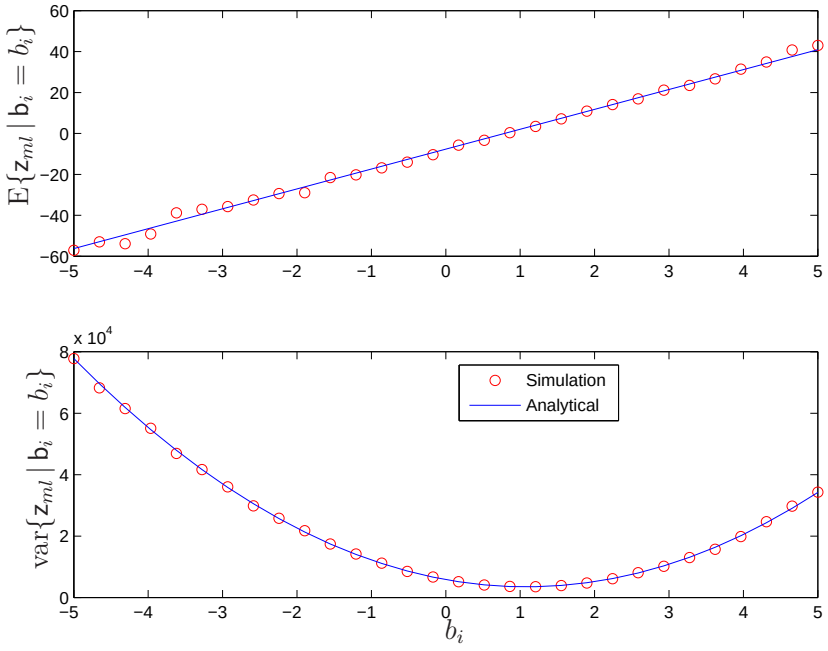


Figure 5.2: The results of a Monte-Carlo simulation used to double-check the derived mean and variance expressions in (5.36) and (5.48). A close agreement between simulated and analytical values is expected given that $10^4$ number of Monte-Carlo averages were used.

message (5.29) from a $(N_b+N_c-1)$-dimensional integral to a one-dimensional integral:

$$\Delta^{\mathsf{b}}_{ml\to i}(t,b_i) \approx \log \int_{z_{ml}} p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}\big(y_{ml} \,\big|\, z_{ml}\big)\mathcal{N}\big(z_{ml}; \mathrm{E}\{\mathsf{z}_{ml} \,|\, \mathsf{b}_i = b_i\}, \mathrm{var}\{\mathsf{z}_{ml} \,|\, \mathsf{b}_i = b_i\}\big) \tag{5.50}$$

$$= H_{ml}\Bigg(\widehat{p}_{i,ml}(t) + b_i\widehat{z}^{(i,*)}_{\to ml}(t), \quad \nu^p_{i,ml}(t) + b_i^2\sum_{j=1}^{N_c}\nu^c_{ml,j}(t)\widehat{z}^{(i,j)2}_{ml} + 2b_i\sum_{j=1}^{N_c}\nu^c_{ml,j}(t)\left[\widehat{z}^{(*,j)}_{\to ml}(t)\widehat{z}^{(i,j)}_{ml} - \widehat{b}_{ml,i}(t)\widehat{z}^{(i,j)2}_{ml}\right]\Bigg) + \mathsf{const}, \tag{5.51}$$

where we have introduced the shorthand notation

$$H_{ml}\big(\widehat{q},\nu^q\big) \triangleq \log \int_z p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml} \,|\, z)\,\mathcal{N}(z;\widehat{q},\nu^q). \tag{5.52}$$

Further approximations to (5.51) will now be made. For this, we first introduce the following $i$-invariant versions of $\widehat{p}_{i,ml}(t)$ and $\nu^p_{i,ml}(t)$:

$$\widehat{p}_{ml}(t) \triangleq \sum_{n=1}^{N} a_{mn}\big(\widehat{\boldsymbol{b}}_{ml}(t)\big)x_{nl}\big(\widehat{\boldsymbol{c}}_{ml}(t)\big) \tag{5.53}$$

$$\nu^p_{ml}(t) \triangleq \sum_{j=1}^{N_c}\nu^c_{ml,j}(t)\widehat{z}^{(*,j)}_{\to ml}(t)^2 + \sum_{k=1}^{N_b}\nu^b_{ml,k}(t)\left[\widehat{z}^{(k,*)}_{\to ml}(t)^2 + \sum_{j=1}^{N_c}\nu^c_{ml,j}(t)\widehat{z}^{(k,j)2}_{ml}\right], \tag{5.54}$$

noting that

$$\widehat{p}_{ml,i}(t) = \widehat{p}_{ml}(t) - \widehat{b}_{ml,i}(t)\widehat{z}^{(i,*)}_{\to ml}(t) \tag{5.55}$$

$$\nu^p_{i,ml}(t) = \nu^p_{ml}(t) + \sum_{j=1}^{N_c}\nu^c_{ml,j}(t)\left[\widehat{b}_{ml,i}(t)^2\widehat{z}^{(i,j)2}_{ml} - 2\widehat{b}_{ml,i}(t)\widehat{z}^{(*,j)}_{\to ml}(t)\widehat{z}^{(i,j)}_{ml}\right] - \nu^b_{ml,i}(t)\left[\widehat{z}^{(i,*)}_{\to ml}(t)^2 + \sum_{j=1}^{N_c}\nu^c_{ml,j}(t)\widehat{z}^{(i,j)2}_{ml}\right]. \tag{5.56}$$

In the sequel, as with $\widehat{p}_{i,ml}(t)$ and $\nu^p_{i,ml}(t)$, we will assume that $\widehat{p}_{ml}(t)$ and $\nu^p_{ml}(t)$ are $O(1)$, since these quantities can be recognized as the mean and variance of a

distribution on the $O(1)$ random variable $\mathsf{z}_{ml}$. Next, we define

$$\widehat{z}_{ml}^{(i,*)}(t) \triangleq \sum_{n=1}^{N} a_{mn}^{(i)} x_{nl}\big(\widehat{\boldsymbol{c}}(t)\big) \tag{5.57}$$

$$\widehat{z}_{ml}^{(*,j)}(t) \triangleq \sum_{n=1}^{N} a_{mn}\big(\widehat{\boldsymbol{b}}(t)\big) x_{nl}^{(j)} \tag{5.58}$$

$$\widehat{z}_{ml}^{(*,*)}(t) \triangleq \sum_{n=1}^{N} a_{mn}\big(\widehat{\boldsymbol{b}}(t)\big) x_{nl}\big(\widehat{\boldsymbol{c}}(t)\big), \tag{5.59}$$

which are versions of (5.42)-(5.43) and $\widehat{p}_{ml}(t)$ evaluated at $\widehat{\boldsymbol{b}}(t)$ and $\widehat{\boldsymbol{c}}(t)$, the means of the SPA-approximated posteriors, rather than at $\widehat{\boldsymbol{b}}_{ml}(t)$ and $\widehat{\boldsymbol{c}}_{ml}(t)$, the means of the SPA messages. As such, the quantities in (5.57)-(5.59) are also $O(1)$. Recalling that $\big(\widehat{c}_{ml,j}(t) - \widehat{c}_j(t)\big)$ is $O(1/N^2)$, an examination of (5.9) under the random affine model shows that $\big(x_{nl}(\widehat{\boldsymbol{c}}_{ml}(t)) - x_{nl}(\widehat{\boldsymbol{c}}(t))\big)$ is $O(1/N)$. Comparing (5.42) and (5.57) and invoking the independence of $\mathbf{b}$ and $\mathbf{c}$, we then conclude that $\big(\widehat{z}_{\to ml}^{(i,*)}(t) - \widehat{z}_{ml}^{(i,*)}(t)\big)$ is $O(1/N)$. A similar line of reasoning shows that $\big(a_{mn}(\widehat{\boldsymbol{b}}_{ml}(t)) - a_{mn}(\widehat{\boldsymbol{b}}(t))\big)$ is $O(1/N^{3/2})$, after which a comparison of (5.43) and (5.58) reveals that $\big(\widehat{z}_{\to ml}^{(*,j)}(t) - \widehat{z}_{ml}^{(*,j)}(t)\big)$ is $O(1/N)$.

Using (5.53) and (5.54), we can now rewrite the $H_{ml}(\cdot)$ term in (5.51) as

$$
\begin{aligned}
&H_{ml}\Bigg(\widehat{p}_{i,ml}(t)+b_i\widehat{z}^{(i,*)}_{\to ml}(t),\quad \nu^p_{i,ml}(t)+b_i^2\sum_{j=1}^{N_c}\nu^c_{ml,j}(t)\widehat{z}^{(i,j)2}_{ml}\\
&\qquad\qquad +2b_i\sum_{j=1}^{N_c}\nu^c_{ml,j}(t)\left[\widehat{z}^{(*,j)}_{\to ml}(t)\widehat{z}^{(i,j)}_{ml}-\widehat{b}_{ml,i}(t)\widehat{z}^{(i,j)2}_{ml}\right]\Bigg)\\
&=H_{ml}\Bigg(\widehat{p}_{ml}(t)+\big(b_i-\widehat{b}_{ml,i}(t)\big)\widehat{z}^{(i,*)}_{\to ml}(t),\quad \nu^p_{ml}(t)+\big(b_i-\widehat{b}_{ml,i}(t)\big)^2\sum_{j=1}^{N_c}\nu^c_{ml,j}(t)\widehat{z}^{(i,j)2}_{ml}\\
&\qquad +2\big(b_i-\widehat{b}_{ml,i}(t)\big)\sum_{j=1}^{N_c}\nu^c_{ml,j}(t)\widehat{z}^{(*,j)}_{\to ml}(t)\widehat{z}^{(i,j)}_{ml}\\
&\qquad -\nu^b_{ml,i}(t)\left[\widehat{z}^{(i,*)}_{\to ml}(t)^2+\sum_{j=1}^{N_c}\nu^c_{ml,j}(t)\widehat{z}^{(i,j)2}_{ml}\right]\Bigg) \qquad (5.60)\\
&=H_{ml}\Bigg(\widehat{p}_{ml}(t)+\big(b_i-\widehat{b}_i(t)\big)\widehat{z}^{(i,*)}_{\to ml}(t)+O(1/N^2),\\
&\qquad \nu^p_{ml}(t)+\big(b_i-\widehat{b}_i(t)\big)^2\sum_{j=1}^{N_c}\nu^c_{ml,j}(t)\widehat{z}^{(i,j)2}_{ml}\\
&\qquad +2\big(b_i-\widehat{b}_i(t)\big)\sum_{j=1}^{N_c}\nu^c_{ml,j}(t)\widehat{z}^{(*,j)}_{ml}(t)\widehat{z}^{(i,j)}_{ml}+O(1/N^2)\Bigg). \qquad (5.61)
\end{aligned}
$$

Next we will perform a Taylor series expansion of the Gaussian-approximated message (5.51) in $b_i$ about $\widehat{b}_i(t)$. For this, we need the first two derivatives of the $H_{ml}(\cdot)$ term w.r.t $b_i$. From (5.61), we find that

$$
\frac{\partial H_{ml}}{\partial b_i}=\widehat{z}^{(i,*)}_{\to ml}(t)H'_{ml}+\left(2\big(b_i-\widehat{b}_i(t)\big)\sum_{j=1}^{N_c}\nu^c_{ml,j}(t)\widehat{z}^{(i,j)2}_{ml}+2\sum_{j=1}^{N_c}\nu^c_{ml,j}(t)\widehat{z}^{(*,j)}_{ml}(t)\widehat{z}^{(i,j)}_{ml}\right)\dot{H}_{ml}, \qquad (5.62)
$$

which implies

$$
\left.\frac{\partial H_{ml}}{\partial b_i}\right|_{b_i=\widehat{b}_i(t)}=\widehat{z}^{(i,*)}_{\to ml}(t)H'_{ml}+\left(2\sum_{j=1}^{N_c}\nu^c_{ml,j}(t)\widehat{z}^{(*,j)}_{ml}(t)\widehat{z}^{(i,j)}_{ml}\right)\dot{H}_{ml}, \qquad (5.63)
$$

and

$$\begin{aligned}\frac{\partial^2 H_{ml}}{\partial b_i^2} &= \widehat{z}_{\to ml}^{(i,*)}(t)^2 H''_{ml} \\ &+ \left(2\big(b_i - \widehat{b}_i(t)\big)\sum_{j=1}^{N_c}\nu_{ml,j}^c(t)\widehat{z}_{ml}^{(i,j)2} + 2\sum_{j=1}^{N_c}\nu_{ml,j}^c(t)\widehat{z}_{ml}^{(*,j)}(t)\widehat{z}_{ml}^{(i,j)}\right)\widehat{z}_{\to ml}^{(i,*)}(t)\dot{H}'_{ml} \\ &+ \left(2\sum_{j=1}^{N_c}\nu_{ml,j}^c(t)\widehat{z}_{ml}^{(i,j)2}\right)\dot{H}_{ml} \\ &+ \left[2\big(b_i - \widehat{b}_i(t)\big)\sum_{j=1}^{N_c}\nu_{ml,j}^c(t)\widehat{z}_{ml}^{(i,j)2} + 2\sum_{j=1}^{N_c}\nu_{ml,j}^c(t)\widehat{z}_{ml}^{(*,j)}(t)\widehat{z}_{ml}^{(i,j)}\right] \\ &\quad\times\left[\widehat{z}_{\to ml}^{(i,*)}(t)\dot{H}'_{ml} + \left(2\big(b_i - \widehat{b}_i(t)\big)\sum_{j=1}^{N_c}\nu_{ml,j}^c(t)\widehat{z}_{ml}^{(i,j)2}\right.\right. \\ &\qquad\left.\left.+ 2\sum_{j=1}^{N_c}\nu_{ml,j}^c(t)\widehat{z}_{ml}^{(*,j)}(t)\widehat{z}_{ml}^{(i,j)}\right)\ddot{H}_{ml}\right] \end{aligned} \tag{5.64}$$

which implies

$$\begin{aligned}\left.\frac{\partial^2 H_{ml}}{\partial\partial b_i^2}\right|_{b_i=\widehat{b}_i(t)} &= \widehat{z}_{\to ml}^{(i,*)}(t)^2 H''_{ml} + \left(4\sum_{j=1}^{N_c}\nu_{ml,j}^c(t)\widehat{z}_{ml}^{(*,j)}(t)\widehat{z}_{ml}^{(i,j)}\right)\widehat{z}_{\to ml}^{(i,*)}(t)\dot{H}'_{ml} \\ &\quad+\left(2\sum_{j=1}^{N_c}\nu_{ml,j}^c(t)\widehat{z}_{ml}^{(i,j)2}\right)\dot{H}_{ml} + \left(2\sum_{j=1}^{N_c}\nu_{ml,j}^c(t)\widehat{z}_{ml}^{(*,j)}(t)\widehat{z}_{ml}^{(i,j)}\right)^2\ddot{H}_{ml},\end{aligned} \tag{5.65}$$

where $H'_{ml}(\cdot,\cdot)$ denotes the derivative w.r.t the first argument and $\dot{H}_{ml}(\cdot,\cdot)$ denotes the derivative w.r.t the second argument. Note that, in the expressions (5.62)-(5.65), the arguments of $H_{ml}$ and its derivatives were suppressed for brevity. The Taylor

series expansion of (5.51) can then be stated as

$$\begin{aligned}
\Delta^{\mathsf{b}}_{ml\to i}(t,b_i) \approx\ & \mathsf{const} + H_{ml}\big(\widehat{p}_{ml}(t) + O(1/N^2), \nu^p_{ml}(t) + O(1/N^2)\big) \\
& + \big(b_i - \widehat{b}_i(t)\big)\Bigg[\widehat{z}^{(i,*)}_{\to ml}(t) H'_{ml}\big(\widehat{p}_{ml}(t) + O(1/N^2), \nu^p_{ml}(t) + O(1/N^2)\big) \\
& \quad + 2\left(\sum_{j=1}^{N_c} \nu^c_{ml,j}(t)\widehat{z}^{(*,j)}_{ml}(t)\widehat{z}^{(i,j)}_{ml}\right) \\
& \quad \times \dot{H}_{ml}\big(\widehat{p}_{ml}(t) + O(1/N^2), \nu^p_{ml}(t) + O(1/N^2)\big)\Bigg] \\
& + \frac{1}{2}\big(b_i - \widehat{b}_i(t)\big)^2\Bigg[\widehat{z}^{(i,*)}_{\to ml}(t)^2 H''_{ml}\big(\widehat{p}_{ml}(t) + O(1/N^2), \nu^p_{ml}(t) + O(1/N^2)\big) \\
& \quad + \left(2\sum_{j=1}^{N_c} \nu^c_{ml,j}(t)\widehat{z}^{(i,j)2}_{ml}\right) \\
& \quad \times \dot{H}_{ml}\big(\widehat{p}_{ml}(t) + O(1/N^2), \nu^p_{ml}(t) + O(1/N^2)\big)\Bigg] \\
& + O(1/N^3), \qquad (5.66)
\end{aligned}$$

where the second and fourth terms in (5.65) were absorbed into the $O(1/N^3)$ term in (5.66) using the facts that

$$\left(4\sum_{j=1}^{N_c} \nu^c_{ml,j}(t)\widehat{z}^{(*,j)}_{ml}(t)\widehat{z}^{(i,j)}_{ml}\right)\widehat{z}^{(i,*)}_{\to ml}(t) = O(1/N) \qquad (5.67)$$

$$\left(2\sum_{j=1}^{N_c} \nu^c_{ml,j}(t)\widehat{z}^{(i,j)2}_{ml}\right) = O(1) \qquad (5.68)$$

$$\left(2\sum_{j=1}^{N_c} \nu^c_{ml,j}(t)\widehat{z}^{(*,j)}_{ml}(t)\widehat{z}^{(i,j)}_{ml}\right)^2 = O(1/N^2). \qquad (5.69)$$

which follow from the random affine parameterization and the $O(1/N^2)$ scaling of $\nu^c_{ml,j}(t)$, as well as from the facts that $\big(b_i - \widehat{b}_i(t)\big)^2$ is $O(1/N^2)$ and the function $H_{ml}$ and its partials are $O(1)$.

Note that the second-order expansion term in (5.66) is $O(1/N^2)$. We will now approximate (5.66) by dropping terms that vanish relative to the latter as $N \to \infty$.

First, we replace $\widehat{z}^{(i,*)}_{\rightarrow ml}(t)$ with $\widehat{z}^{(i,*)}_{ml}(t)$ in the quadratic term in (5.66), since $\big(\widehat{z}^{(i,*)}_{\rightarrow ml}(t) - \widehat{z}^{(i,*)}_{ml}(t)\big)$ is $O(1/N)$, which gets reduced to $O(1/N^3)$ via scaling by $\big(b_i - \widehat{b}_i(t)\big)^2$. Note that we cannot make a similar replacement in the linear term in (5.66), because the $\big(b_i - \widehat{b}_i(t)\big)$ scaling is not enough to render the difference negligible. Next, we replace $\nu^c_{ml,j}(t)$ with $\nu^c_j(t)$ throughout (5.66), since the difference is $O(1/N^3)$. Finally, as established in [8], the $O(1/N^2)$ perturbations inside the $H_{ml}$ derivatives can be dropped because they have an $O(1/N^3)$ effect on the overall message. With these approximations, and absorbing $b_i$-invariant terms into the const, we obtain

$$\begin{aligned}
\Delta^{\mathsf{b}}_{ml\rightarrow i}(t, b_i) &\approx \mathsf{const} \\
&+ \Bigg[\widehat{s}_{ml}(t)\widehat{z}^{(i,*)}_{\rightarrow ml}(t) + \big(\widehat{s}^2_{ml}(t) - \nu^s_{ml}(t)\big)\sum_{j=1}^{N_c}\nu^c_j(t)\widehat{z}^{(i,j)}_{ml}\Big(\widehat{z}^{(*,j)}_{ml}(t) - \widehat{b}_i(t)\widehat{z}^{(i,j)}_{ml}\Big) \\
&\quad + \nu^s_{ml}(t)\widehat{b}_i(t)\widehat{z}^{(i,*)}_{ml}(t)^2\Bigg] b_i \\
&- \frac{1}{2}\Bigg[\nu^s_{ml}(t)\widehat{z}^{(i,*)}_{ml}(t)^2 - \big(\widehat{s}^2_{ml}(t) - \nu^s_{ml}(t)\big)\sum_{j=1}^{N_c}\nu^c_j(t)\widehat{z}^{(i,j)2}_{ml}\Bigg] b_i^2, \qquad (5.70)
\end{aligned}$$

where we used

$$\widehat{s}_{ml}(t) \triangleq H'_{ml}\big(\widehat{p}_{ml}(t), \nu^p_{ml}(t)\big) \qquad (5.71)$$

$$\nu^s_{ml}(t) \triangleq -H''_{ml}\big(\widehat{p}_{ml}(t), \nu^p_{ml}(t)\big) \qquad (5.72)$$

and the following relationship established in [8]:

$$\dot{H}_{ml}\big(q, \nu^q\big) = \frac{1}{2}\Big[H'_{ml}\big(q, \nu^q\big)^2 + H''_{ml}\big(q, \nu^q\big)\Big]. \qquad (5.73)$$

Note that, since the message (5.70) is quadratic, we have essentially made a Gaussian approximation to the pdf $\frac{1}{C}\exp(\Delta^{\mathsf{b}}_{ml\rightarrow i}(t, .))$. Also note that, since the function $H_{ml}(\cdot)$ and its partials are $O(1)$, and since all the arguments of these functions are $O(1)$, we conclude that (5.71) and (5.72) are $O(1)$ as well.

Furthermore, the derivation in [8, Appendix A] shows that (5.71)-(5.72) can be rewritten as

$$\widehat{s}_{ml}(t) = \frac{1}{\nu^p_{ml}(t)}\big(\widehat{z}_{ml}(t) - \widehat{p}_{ml}(t)\big) \tag{5.74}$$

$$\nu^s_{ml}(t) = \frac{1}{\nu^p_{ml}(t)}\left(1 - \frac{\nu^z_{ml}(t)}{\nu^p_{ml}(t)}\right), \tag{5.75}$$

using the conditional mean and variance

$$\widehat{z}_{ml}(t) \triangleq \mathrm{E}\{\mathsf{z}_{ml} \,|\, \mathsf{p}_{ml}{=}\widehat{p}_{ml}(t); \nu^p_{ml}(t)\} \tag{5.76}$$

$$\nu^z_{ml}(t) \triangleq \mathrm{var}\{\mathsf{z}_{ml} \,|\, \mathsf{p}_{ml}{=}\widehat{p}_{ml}(t); \nu^p_{ml}(t)\}, \tag{5.77}$$

computed according to the (conditional) pdf

$$\begin{aligned} &p_{\mathsf{z}_{ml}|\mathsf{p}_{ml}}\big(z_{ml} \,|\, \widehat{p}_{ml}(t); \nu^p_{ml}(t)\big) \\ &\quad\triangleq \tfrac{1}{C}\, p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml} \,|\, z_{ml})\, \mathcal{N}\big(z_{ml}; \widehat{p}_{ml}(t), \nu^p_{ml}(t)\big), \end{aligned} \tag{5.78}$$

where here $C = \int_z p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml} \,|\, z)\mathcal{N}\big(z; \widehat{p}_{ml}(t), \nu^p_{ml}(t)\big)$. In fact, (5.78) is P-BiG-AMP's iteration-$t$ approximation to the true marginal posterior $p_{\mathsf{z}_{ml}|\mathsf{Y}}(z_{ml}|\boldsymbol{Y})$. We note that (5.78) can also be interpreted as the (exact) posterior pdf for $\mathsf{z}_{ml}$ given the likelihood $p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml}|\cdot)$ from (5.3) and the prior $\mathsf{z}_{ml} \sim \mathcal{N}\big(\widehat{p}_{ml}(t), \nu^p_{ml}(t)\big)$ that is implicitly adopted by iteration-$t$ P-BiG-AMP.

### 5.5.2 SPA message from node $p_{\mathbf{y}_{ml}|\mathbf{z}_{ml}}$ to node $\mathbf{c}_j$

Since the bilinear form $\mathsf{z}_{ml} = \sum_{n=1}^{N}\sum_{i=0}^{N_b} a^{(i)}_{mn}\mathsf{b}_i \sum_{j=0}^{N_c} x^{(j)}_{nl}\mathsf{c}_j$ implies a symmetry between $\mathsf{b}_i$ and $\mathsf{c}_j$, the derivation of $\Delta^{\mathsf{c}}_{ml\to j}(t,\cdot)$ closely follows the derivation of $\Delta^{\mathsf{b}}_{ml\to i}(t,\cdot)$

from Section 5.5.1. Following the same process, we begin with

$$\Delta^{\mathsf{c}}_{ml\to j}(t,c_j) = \log \int_{\{b_r\}_{r=1}^{N_b},\{c_k\}_{k\neq j}} p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}\Big(y_{ml}\,\Big|\,\overbrace{\sum_{n=1}^{N} a_{mn}(\boldsymbol{b})x_{nl}(\boldsymbol{c})}^{z_{ml}}\Big) \times \prod_{r=1}^{N_b}\exp\Big(\Delta^{\mathsf{b}}_{ml\leftarrow r}(t,b_r)\Big)\prod_{k\neq j}\exp\Big(\Delta^{\mathsf{c}}_{ml\leftarrow k}(t,c_k)\Big) + \mathsf{const} \tag{5.79}$$

and apply the CLT in order to treat $\mathsf{z}_{ml}|\mathsf{b}_{ml}=b_i$, i.e., the random variable associated with the $z_{ml}$ identified in (5.79), as Gaussian in the large-system limit. Using the same methods as before, the conditional mean becomes

$$\mathrm{E}\{\mathsf{z}_{ml}\,|\,\mathsf{c}_j=c_j\} = \widehat{p}_{j,ml}(t) + c_j\widehat{z}^{(*,j)}_{\to ml}(t), \tag{5.80}$$

where, with an abuse of notation,[44] we employed the definition

$$\widehat{p}_{j,ml}(t) \triangleq \sum_{n=1}^{N} a_{mn}\big(\widehat{\boldsymbol{b}}_{ml}(t)\big)x_{nl}\big(\widehat{\boldsymbol{c}}_{ml}(t)\big) - \widehat{c}_{ml,j}(t)\sum_{n=1}^{N} a_{mn}\big(\widehat{\boldsymbol{b}}_{ml}(t)\big)x^{(j)}_{nl} \tag{5.81}$$

$$= \widehat{p}_{ml}(t) - \widehat{c}_{ml,j}(t)\widehat{z}^{(*,j)}_{\to ml}(t). \tag{5.82}$$

Similarly, the conditional variance is

$$\mathrm{var}\{\mathsf{z}_{ml}\,|\,\mathsf{c}_j=c_j\} = \nu^{p}_{j,ml}(t) + c_j^2\sum_{i=1}^{N_b}\nu^{b}_{ml,i}(t)\widehat{z}^{(i,j)2}_{ml} + 2c_j\sum_{i=1}^{N_b}\nu^{b}_{ml,i}(t)\left[\widehat{z}^{(i,*)}_{\to ml}(t)\widehat{z}^{(i,j)}_{ml} - \widehat{c}_{ml,j}(t)\widehat{z}^{(i,j)2}_{ml}\right], \tag{5.83}$$

where, again with an abuse of notation, we employed the definition

$$\nu^{p}_{j,ml}(t) \triangleq \sum_{i=1}^{N_b}\nu^{b}_{ml,i}(t)\left[\widehat{z}^{(i,*)}_{\to ml}(t)^2 + \widehat{c}_{ml,j}(t)^2\widehat{z}^{(i,j)2}_{ml} - 2\widehat{c}_{ml,j}(t)\widehat{z}^{(i,*)}_{\to ml}(t)\widehat{z}^{(i,j)}_{ml}\right] + \sum_{k\neq j}\nu^{c}_{ml,k}(t)\left[\widehat{z}^{(*,k)}_{\to ml}(t)^2 + \sum_{i=1}^{N_b}\nu^{c}_{ml,k}(t)\widehat{z}^{(i,k)}_{ml}(t)^2\right]. \tag{5.84}$$

[44]Notice that $\widehat{p}_{i,ml}(t)$ and $\widehat{p}_{j,ml}(t)$ have different functional forms. However, both reduce to $\widehat{p}_{ml}(t)$ from (5.53) when the dependance on destination (i.e., $i$ or $j$) is removed.

The conditional-Gaussian approximation reduces the high-dimensional integral in (5.79) to a one-dimensional integral, so that

$$\begin{aligned}
&\Delta^{\mathsf{c}}_{ml\to j}(t,c_j) \qquad (5.85)\\
&\approx \log \int_{z_{ml}} p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}\big(y_{ml}\,\big|\,z_{ml}\big)\mathcal{N}\big(z_{ml};\mathrm{E}\{\mathsf{z}_{ml}\,|\,\mathsf{c}_j=c_j\},\mathrm{var}\{\mathsf{z}_{ml}\,|\,\mathsf{c}_j=c_j\}\big)\\
&= H_{ml}\Bigg(\widehat{p}_{j,ml}(t)+c_j\widehat{z}^{(*,j)}_{\to ml}(t),\quad \nu^p_{j,ml}(t)+c_j^2\sum_{i=1}^{N_b}\nu^b_{ml,i}(t)\widehat{z}^{(i,j)2}_{ml}\\
&\qquad\qquad +2c_j\sum_{i=1}^{N_b}\nu^b_{ml,i}(t)\left[\widehat{z}^{(i,*)}_{\to ml}(t)\widehat{z}^{(i,j)}_{ml}-\widehat{c}_{ml,j}(t)\widehat{z}^{(i,j)2}_{ml}\right]\Bigg)+\mathsf{const} \qquad (5.86)\\
&= \mathsf{const}+H_{ml}\Bigg(\widehat{p}_{ml}(t)+\big(c_j-\widehat{c}_j(t)\big)\widehat{z}^{(*,j)}_{\to ml}(t)+O(1/N^2),\\
&\qquad\qquad \nu^p_{ml}(t)+\big(c_j-\widehat{c}_j(t)\big)^2\sum_{i=1}^{N_b}\nu^b_{ml,i}(t)\widehat{z}^{(i,j)2}_{ml}\\
&\qquad\qquad\qquad +2\big(c_j-\widehat{c}_j(t)\big)\sum_{i=1}^{N_b}\nu^b_{ml,i}(t)\widehat{z}^{(i,*)}_{ml}(t)\widehat{z}^{(i,j)}_{ml}+O(1/N^2)\Bigg), \qquad (5.87)
\end{aligned}$$

where (5.87) uses the same line of reasoning as (5.61). Taking a Taylor series expansion of (5.87) in $c_j$ about $\widehat{c}_j(t)$, we obtain

$$\begin{aligned}
\Delta^{\mathsf{c}}_{ml\to j}(t,c_j) \approx\ & \mathsf{const}+H_{ml}\big(\widehat{p}_{ml}(t)+O(1/N^2),\nu^p_{ml}(t)+O(1/N^2)\big)\\
&+\big(c_j-\widehat{c}_j(t)\big)\Bigg[\widehat{z}^{(*,j)}_{\to ml}(t)H'_{ml}\big(\widehat{p}_{ml}(t)+O(1/N^2),\nu^p_{ml}(t)+O(1/N^2)\big)\\
&\quad+2\left(\sum_{i=1}^{N_b}\nu^b_{ml,i}(t)\widehat{z}^{(i,*)}_{ml}(t)\widehat{z}^{(i,j)}_{ml}\right)\\
&\qquad\times\dot{H}_{ml}\big(\widehat{p}_{ml}(t)+O(1/N^2),\nu^p_{ml}(t)+O(1/N^2)\big)\Bigg]\\
&+\frac{1}{2}\Big(c_j-\widehat{c}_j(t)\Big)^2\Bigg[\widehat{z}^{(*,j)}_{\to ml}(t)^2H''_{ml}\big(\widehat{p}_{ml}(t)+O(1/N^2),\nu^p_{ml}(t)+O(1/N^2)\big)\\
&\quad+\left(2\sum_{i=1}^{N_b}\nu^b_{ml,i}(t)\widehat{z}^{(i,j)2}_{ml}\right)\dot{H}_{ml}\big(\widehat{p}_{ml}(t)+O(1/N^2),\nu^p_{ml}(t)+O(1/N^2)\big)\Bigg]\\
&+O(1/N^3) \qquad (5.88)
\end{aligned}$$

using arguments similar to those used for (5.66). Finally, as in (5.70), we neglect terms that vanish relative to the quadratic term in (5.88), yielding

$$\begin{aligned}
&\Delta^{\mathsf{c}}_{ml\to j}(t,c_j) \approx \mathsf{const} \\
&\quad + \Bigg[\widehat{s}_{ml}(t)\widehat{z}^{(*,j)}_{\to ml}(t) + \big(\widehat{s}^2_{ml}(t) - \nu^s_{ml}(t)\big)\sum_{i=1}^{N_b}\nu^b_i(t)\widehat{z}^{(i,j)}_{ml}\Big(\widehat{z}^{(i,*)}_{ml}(t) - \widehat{c}_j(t)\widehat{z}^{(i,j)}_{ml}\Big) \\
&\qquad + \nu^s_{ml}(t)\widehat{c}_j(t)\widehat{z}^{(*,j)}_{ml}(t)^2\Bigg]c_j \\
&\quad - \frac{1}{2}\Bigg[\nu^s_{ml}(t)\widehat{z}^{(*,j)}_{ml}(t)^2 - \big(\widehat{s}^2_{ml}(t) - \nu^s_{ml}(t)\big)\sum_{i=1}^{N_b}\nu^b_i(t)\widehat{z}^{(i,j)2}_{ml}\Bigg]c_j^2. \qquad (5.89)
\end{aligned}$$

At this point, we have obtained quadratic approximations to all the messages flowing out from the $p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}$ nodes. We now turn to approximating the messages flowing out of the variable nodes.

### 5.5.3 SPA message from node $\mathbf{c}_j$ to $p_{\mathbf{y}_{ml}|\mathbf{z}_{ml}}$

We start with (5.26) and plug in the approximation for $\Delta^{\mathsf{c}}_{ml\to j}(t,c_j)$ in (5.89) to obtain

$$\begin{aligned}
&\Delta^{\mathsf{c}}_{ml\leftarrow j}(t+1,c_j) \\
&\quad \approx \mathsf{const} + \log p_{\mathsf{c}_j}(c_j) \\
&\quad + \sum_{(r,k)\neq(m,l)}\Bigg(\Bigg[\widehat{s}_{rk}(t)\widehat{z}^{(*,j)}_{\to rk}(t) + \big(\widehat{s}^2_{rk}(t) - \nu^s_{rk}(t)\big)\sum_{i=1}^{N_b}\nu^b_i(t)\widehat{z}^{(i,j)}_{rk}\Big(\widehat{z}^{(i,*)}_{rk}(t) - \widehat{c}_j(t)\widehat{z}^{(i,j)}_{rk}\Big) \\
&\qquad + \nu^s_{rk}(t)\widehat{c}_j(t)\widehat{z}^{(*,j)}_{rk}(t)^2\Bigg]c_j - \frac{1}{2}\Bigg[\nu^s_{rk}(t)\widehat{z}^{(*,j)}_{rk}(t)^2 - \big(\widehat{s}^2_{rk}(t) - \nu^s_{rk}(t)\big)\sum_{i=1}^{N_b}\nu^b_i(t)\widehat{z}^{(i,j)2}_{rk}\Bigg]c_j^2\Bigg) \qquad (5.90) \\
&\quad = \mathsf{const} + \log p_{\mathsf{c}_j}(c_j) - \frac{1}{2\nu^r_{ml,j}(t)}\big(c_j - \widehat{r}_{ml,j}(t)\big)^2 \qquad (5.91) \\
&\quad = \mathsf{const} + \log\Big(p_{\mathsf{c}_j}(c_j)\mathcal{N}\big(c_j;\widehat{r}_{ml,j}(t),\nu^r_{ml,j}(t)\big)\Big) \qquad (5.92)
\end{aligned}$$

where

$$\nu^r_{ml,j}(t) \triangleq \left[ \sum_{(r,k)\neq(m,l)} \left( \nu^s_{rk}(t)\widehat{z}^{(*,j)}_{rk}(t)^2 - \left(\widehat{s}^2_{rk}(t) - \nu^s_{rk}(t)\right) \sum_{i=1}^{N_b} \nu^b_i(t)\widehat{z}^{(i,j)2}_{rk} \right) \right]^{-1} \tag{5.93}$$

$$\widehat{r}_{ml,j}(t) \triangleq \widehat{c}_j(t) + \nu^r_{ml,j}(t) \sum_{(r,k)\neq(m,l)} \left( \left(\widehat{s}^2_{rk}(t) - \nu^s_{rk}(t)\right) \sum_{i=1}^{N_b} \nu^b_i(t)\widehat{z}^{(i,j)}_{rk}\widehat{z}^{(i,*)}_{rk}(t) + \widehat{s}_{rk}(t)\widehat{z}^{(*,j)}_{\to rk}(t) \right). \tag{5.94}$$

Since $\nu^r_{ml,j}(t)$ is the reciprocal of a sum of $ML$ terms of $O(1)$, we conclude that it is $O(1/N^2)$. Given this and the scalings from Table 5.2, we see that $\widehat{r}_{ml,j}(t)$ is $O(1/N)$. Since $\widehat{r}_{ml,j}(t)$ can be interpreted as an estimate of $\mathsf{c}_j$, this scaling is anticipated.

The mean and variance of the pdf associated with the $\Delta^{\mathsf{c}}_{ml\leftarrow j}(t+1, c_j)$ message approximation from (5.92) are

$$\widehat{c}_{ml,j}(t+1) \triangleq \underbrace{\frac{1}{K}\int_c c\, p_{\mathsf{c}_j}(c)\mathcal{N}\left(c; \widehat{r}_{ml,j}(t), \nu^r_{ml,j}(t)\right)}_{\triangleq\, g_{\mathsf{c}_j}(\widehat{r}_{ml,j}(t), \nu^r_{ml,j}(t))} \tag{5.95}$$

$$\nu^c_{ml,j}(t+1) \triangleq \underbrace{\frac{1}{K}\int_{\mathsf{c}} \left|c - \widehat{c}_{ml,j}(t+1)\right|^2 p_{\mathsf{c}_j}(c)\mathcal{N}\left(c; \widehat{r}_{ml,j}(t), \nu^r_{ml,j}(t)\right)}_{\nu^r_{ml,j}(t)\, g'_{\mathsf{c}_j}(\widehat{r}_{ml,j}(t), \nu^r_{ml,j}(t))} \tag{5.96}$$

where $K = \int_c p_{\mathsf{c}_j}(c)\mathcal{N}\left(c; \widehat{r}_{ml,j}(t), \nu^r_{ml,j}(t)\right)$ here and $g'_{\mathsf{c}_j}$ denotes the derivative of $g_{\mathsf{c}_j}$ with respect to the first argument. The fact that (5.95) and (5.96) are related through a derivative was shown in [6].

Next we develop mean and variance approximations that do not depend on the destination node $ml$. For this, we introduce $ml$-invariant versions of $\widehat{r}_{ml,j}(t)$ and

$\nu^r_{ml,j}(t)$:

$$\nu^r_j(t) \triangleq \left[\sum_{m,l}\left(\nu^s_{ml}(t)\widehat{z}^{(*,j)}_{ml}(t)^2 - \left(\widehat{s}^2_{ml}(t) - \nu^s_{ml}(t)\right)\sum_{i=1}^{N_b}\nu^b_i(t)\widehat{z}^{(i,j)2}_{ml}\right)\right]^{-1} \tag{5.97}$$

$$\widehat{r}_j(t) \triangleq \widehat{c}_j(t) + \nu^r_j(t)\sum_{m,l}\left(\left(\widehat{s}^2_{ml}(t) - \nu^s_{ml}(t)\right)\sum_{i=1}^{N_b}\nu^b_i(t)\widehat{z}^{(i,j)}_{ml}\widehat{z}^{(i,*)}_{ml}(t) + \widehat{s}_{ml}(t)\widehat{z}^{(*,j)}_{\to ml}(t)\right). \tag{5.98}$$

Comparing (5.93)-(5.94) and (5.97)-(5.98) reveals that $\left(\nu^r_{ml,j}(t) - \nu^r_j(t)\right)$ scales as $O(1/N^4)$ and that $\widehat{r}_{ml,j}(t) = \widehat{r}_j(t) - \nu^r_j(t)\widehat{s}_{ml}(t)\widehat{z}^{(*,j)}_{ml}(t) + O(1/N^3)$, and thus (5.95) implies

$$\widehat{c}_{ml,j}(t+1) = g_{\mathsf{c}_j}\Big(\widehat{r}_j(t) - \nu^r_j(t)\widehat{s}_{ml}(t)\widehat{z}^{(*,j)}_{ml}(t) + O(1/N^3),\ \nu^r_j(t) + O(1/N^4)\Big) \tag{5.99}$$

$$= g_{\mathsf{c}_j}\Big(\widehat{r}_j(t) - \nu^r_j(t)\widehat{s}_{ml}(t)\widehat{z}^{(*,j)}_{ml}(t),\ \nu^r_j(t)\Big) + O(1/N^3) \tag{5.100}$$

$$= g_{\mathsf{c}_j}\big(\widehat{r}_j(t), \nu^r_j(t)\big) - \nu^r_j(t)g'_{\mathsf{c}_j}\big(\widehat{r}_j(t), \nu^r_j(t)\big)\widehat{s}_{ml}(t)\widehat{z}^{(*,j)}_{ml}(t) + O(1/N^3) \tag{5.101}$$

$$= \widehat{c}_j(t+1) - \widehat{s}_{ml}(t)\widehat{z}^{(*,j)}_{ml}(t)\nu^c_j(t+1) + O(1/N^3), \tag{5.102}$$

where (5.100) follows by taking Taylor series expansions of (5.99) about the perturbations to the arguments; (5.101) follows by taking a Taylor series expansion of (5.100) in the first argument about the point $\widehat{r}_j(t)$; and (5.102) follows from the definitions

$$\widehat{c}_j(t+1) \triangleq g_{\mathsf{c}_j}\big(\widehat{r}_j(t), \nu^r_j(t)\big) \tag{5.103}$$

$$\nu^c_j(t+1) \triangleq \nu^r_j(t)g'_{\mathsf{c}_j}\big(\widehat{r}_j(t), \nu^r_j(t)\big). \tag{5.104}$$

Note that (5.102) is consistent with our earlier assumption that $\left(\widehat{c}_{ml,j}(t+1) - \widehat{c}_j(t+1)\right)$ is $O(1/N^2)$.

### 5.5.4 SPA message from node $\mathbf{b}_i$ to $p_{\mathbf{y}_{ml}|\mathbf{z}_{ml}}$

Once again, due to symmetry, the derivation for $\Delta^{\mathsf{b}}_{ml\leftarrow i}(t+1, b_i)$ closely parallels that for $\Delta^{\mathsf{c}}_{ml\leftarrow j}(t+1, c_j)$. Plugging approximation (5.70) into (5.25), we obtain

$$\begin{aligned}
&\Delta^{\mathsf{b}}_{ml\leftarrow i}(t+1, b_i) \\
&\quad \approx \mathsf{const} + \log p_{\mathsf{b}_i}(b_i) \\
&\quad + \sum_{(r,k)\neq(m,l)} \Bigg( \Bigg[ \widehat{s}_{rk}(t)\widehat{z}^{(i,*)}_{\rightarrow rk}(t) + \big(\widehat{s}^2_{rk}(t) - \nu^s_{rk}(t)\big) \sum_{j=1}^{N_c} \nu^c_j(t) \widehat{z}^{(i,j)}_{rk} \Big( \widehat{z}^{(*,j)}_{rk}(t) - \widehat{b}_i(t)\widehat{z}^{(i,j)}_{rk} \Big) \\
&\qquad + \nu^s_{rk}(t)\widehat{b}_i(t)\widehat{z}^{(i,*)}_{rk}(t)^2 \Bigg] b_i - \frac{1}{2} \Bigg[ \nu^s_{rk}(t)\widehat{z}^{(i,*)}_{rk}(t)^2 - \big(\widehat{s}^2_{rk}(t) - \nu^s_{rk}(t)\big) \sum_{j=1}^{N_c} \nu^c_j(t)\widehat{z}^{(i,j)2}_{rk} \Bigg] b_i^2 \Bigg) \qquad (5.105) \\
&\quad = \mathsf{const} + \log \Big( p_{\mathsf{c}_i}(b_i) \mathcal{N}\big(b_i; \widehat{q}_{ml,i}(t), \nu^q_{ml,i}(t)\big) \Big) \qquad (5.106)
\end{aligned}$$

where

$$\nu^q_{ml,i}(t) \triangleq \Bigg[ \sum_{(r,k)\neq(m,l)} \Bigg( \nu^s_{rk}(t)\widehat{z}^{(i,*)}_{rk}(t)^2 - \big(\widehat{s}^2_{rk}(t) - \nu^s_{rk}(t)\big) \sum_{j=1}^{N_c} \nu^c_j(t)\widehat{z}^{(i,j)2}_{rk} \Bigg) \Bigg]^{-1} \qquad (5.107)$$

$$\begin{aligned}
\widehat{q}_{ml,i}(t) &\triangleq \widehat{b}_i(t) \\
&\quad + \nu^q_{ml,i}(t) \sum_{(r,k)\neq(m,l)} \Bigg( \big(\widehat{s}^2_{rk}(t) - \nu^s_{rk}(t)\big) \sum_{j=1}^{N_c} \nu^c_j(t) \widehat{z}^{(i,j)}_{rk} \widehat{z}^{(*,j)}_{rk}(t) + \widehat{s}_{rk}(t)\widehat{z}^{(i,*)}_{\rightarrow rk}(t) \Bigg). \qquad (5.108)
\end{aligned}$$

Due to the symmetry and previously established scalings, the $q$ variables follow the same scaling as the $r$ variables, as shown in Table 5.2.

The mean and variance of the pdf associated with the $\Delta^{\mathsf{b}}_{ml\leftarrow i}(t{+}1, b_i)$ approximation in (5.106) are

$$\widehat{b}_{ml,i}(t{+}1) \triangleq \underbrace{\frac{1}{K}\int_b b\, p_{\mathsf{b}_i}(b)\mathcal{N}\big(b; \widehat{q}_{ml,i}(t), \nu^q_{ml,i}(t)\big)}_{\triangleq\, g_{\mathsf{b}_i}(\widehat{q}_{ml,i}(t), \nu^q_{ml,i}(t))} \tag{5.109}$$

$$\nu^b_{ml,i}(t{+}1) \triangleq \underbrace{\frac{1}{K}\int_{\mathsf{b}} \big|b - \widehat{b}_{ml,i}(t{+}1)\big|^2 p_{\mathsf{b}_i}(b)\mathcal{N}\big(b; \widehat{q}_{ml,i}(t), \nu^q_{ml,i}(t)\big)}_{\nu^q_{ml,i}(t)\, g'_{\mathsf{b}_i}(\widehat{q}_{ml,i}(t), \nu^q_{ml,i}(t))} \tag{5.110}$$

where $K = \int_b p_{\mathsf{b}_i}(b)\mathcal{N}\big(b; \widehat{q}_{ml,i}(t), \nu^q_{ml,i}(t)\big)$ here and $g'_{\mathsf{b}_i}$ denotes the derivative of $g_{\mathsf{b}_i}$ with respect to the first argument. Following the same procedure as before, we define the $ml$-invariant quantities

$$\nu^q_i(t) \triangleq \left[\sum_{m,l}\left(\nu^s_{ml}(t)\widehat{z}^{(i,*)}_{ml}(t)^2 - \big(\widehat{s}^2_{ml}(t) - \nu^s_{ml}(t)\big)\sum_{j=1}^{N_c}\nu^c_j(t)\widehat{z}^{(i,j)2}_{ml}\right)\right]^{-1} \tag{5.111}$$

$$\widehat{q}_i(t) \triangleq \widehat{b}_i(t) + \nu^q_i(t)\sum_{m,l}\left(\big(\widehat{s}^2_{ml}(t) - \nu^s_{ml}(t)\big)\sum_{j=1}^{N_c}\nu^c_j(t)\widehat{z}^{(i,j)}_{ml}\widehat{z}^{(*,j)}_{ml}(t) + \widehat{s}_{ml}(t)\widehat{z}^{(i,*)}_{\to ml}(t)\right) \tag{5.112}$$

and perform several Taylor series expansions, finally dropping terms that vanish in the large system limit, to obtain

$$\widehat{b}_{ml,i}(t{+}1) = g_{\mathsf{b}_i}\Big(\widehat{q}_i(t) - \nu^q_i(t)\widehat{s}_{ml}(t)\widehat{z}^{(i,*)}_{ml}(t) + O(1/N^3),\ \nu^q_i(t) + O(1/N^4)\Big) \tag{5.113}$$

$$= g_{\mathsf{b}_i}\Big(\widehat{q}_i(t) - \nu^q_i(t)\widehat{s}_{ml}(t)\widehat{z}^{(i,*)}_{ml}(t),\ \nu^q_i(t)\Big) + O(1/N^3) \tag{5.114}$$

$$= g_{\mathsf{b}_i}\big(\widehat{q}_i(t), \nu^q_i(t)\big) - \nu^q_i(t)g'_{\mathsf{b}_i}\big(\widehat{q}_i(t), \nu^q_i(t)\big)\widehat{s}_{ml}(t)\widehat{z}^{(i,*)}_{ml}(t) + O(1/N^3) \tag{5.115}$$

$$= \widehat{b}_i(t{+}1) - \widehat{s}_{ml}(t)\widehat{z}^{(i,*)}_{ml}(t)\nu^b_i(t{+}1) + O(1/N^3), \tag{5.116}$$

where we have used the definitions

$$\widehat{b}_i(t{+}1) \triangleq g_{\mathsf{b}_i}\big(\widehat{q}_i(t), \nu^q_i(t)\big) \tag{5.117}$$

$$\nu^b_i(t{+}1) \triangleq \nu^q_i(t)g'_{\mathsf{b}_i}\big(\widehat{q}_i(t), \nu^q_i(t)\big). \tag{5.118}$$

### 5.5.5 Closing the loop

To complete the derivation of P-BiG-AMP, we use (5.102) and (5.116) to eliminate the dependence on $ml$ in the $\mathsf{b}_i$ and $\mathsf{c}_j$ estimates and to eliminate the dependence on $i$ and $j$ in the $\mathsf{z}_{ml}$ estimates. First, applying (5.102) to (5.9) evaluated at $\boldsymbol{c} = \widehat{\boldsymbol{c}}_{ml}(t)$ and $\widehat{\boldsymbol{c}} = \widehat{\boldsymbol{c}}(t)$ yields

$$x_{nl}\big(\widehat{\boldsymbol{c}}_{ml}(t)\big) = x_{nl}\big(\widehat{\boldsymbol{c}}(t)\big) - \widehat{s}_{ml}(t-1)\sum_{j=1}^{N_c}\widehat{z}_{ml}^{(*,j)}(t-1)\nu_j^c(t)x_{nl}^{(j)} + O(1/N^2), \tag{5.119}$$

and applying (5.116) to (5.8) yields

$$a_{mn}\big(\widehat{\boldsymbol{b}}_{ml}(t)\big) = a_{mn}\big(\widehat{\boldsymbol{b}}(t)\big) - \widehat{s}_{ml}(t-1)\sum_{i=1}^{N_b}\widehat{z}_{ml}^{(i,*)}(t-1)\nu_i^b(t)a_{mn}^{(i)} + O(1/N^{2.5}). \tag{5.120}$$

Equations (5.119)-(5.120) can be used to write $\widehat{p}_{ml}(t)$ from (5.53) as

$$\begin{aligned}
\widehat{p}_{ml}(t) &= \sum_{n=1}^{N}\left(a_{mn}\big(\widehat{\boldsymbol{b}}(t)\big) - \widehat{s}_{ml}(t-1)\sum_{i=1}^{N_b}\widehat{z}_{ml}^{(i,*)}(t-1)\nu_i^b(t)a_{mn}^{(i)}\right) \\
&\quad\times\left(x_{nl}\big(\widehat{\boldsymbol{c}}(t)\big) - \widehat{s}_{ml}(t-1)\sum_{j=1}^{N_c}\widehat{z}_{ml}^{(*,j)}(t-1)\nu_j^c(t)x_{nl}^{(j)}\right) + O(1/N^2) \qquad (5.121)\\
&= \widehat{z}_{ml}^{(*,*)}(t) - \widehat{s}_{ml}(t-1)\left(\sum_{i=1}^{N_b}\widehat{z}_{ml}^{(i,*)}(t-1)\widehat{z}_{ml}^{(i,*)}(t)\nu_i^b(t) + \sum_{j=1}^{N_c}\widehat{z}_{ml}^{(*,j)}(t-1)\widehat{z}_{ml}^{(*,j)}(t)\nu_j^c(t)\right) \\
&\quad + \widehat{s}_{ml}^2(t-1)\left(\sum_{i=1}^{N_b}\sum_{j=1}^{N_c}\nu_i^b(t)\nu_j^c(t)\widehat{z}_{ml}^{(i,j)}\widehat{z}_{ml}^{(i,*)}(t-1)\widehat{z}_{ml}^{(*,j)}(t-1)\right) + O(1/N^2) \qquad (5.122)\\
&\approx \widehat{z}_{ml}^{(*,*)}(t) - \widehat{s}_{ml}(t-1)\left(\sum_{i=1}^{N_b}\widehat{z}_{ml}^{(i,*)}(t-1)\widehat{z}_{ml}^{(i,*)}(t)\nu_i^b(t) + \sum_{j=1}^{N_c}\widehat{z}_{ml}^{(*,j)}(t-1)\widehat{z}_{ml}^{(*,j)}(t)\nu_j^c(t)\right), \qquad (5.123)
\end{aligned}$$

where the final step follows by neglecting terms that vanish in the large system limit relative to the remaining $O(1)$ terms. Although not justified by the large-system

limit, we will also apply the approximations

$$\sum_{i=1}^{N_b} \widehat{z}_{ml}^{(i,*)}(t-1)\widehat{z}_{ml}^{(i,*)}(t)\nu_i^b(t) \approx \sum_{i=1}^{N_b} \widehat{z}_{ml}^{(i,*)}(t)^2\nu_i^b(t) \tag{5.124}$$

$$\sum_{j=1}^{N_c} \widehat{z}_{ml}^{(*,j)}(t-1)\widehat{z}_{ml}^{(*,j)}(t)\nu_j^c(t) \approx \sum_{j=1}^{N_c} \widehat{z}_{ml}^{(*,j)}(t)^2\nu_j^c(t) \tag{5.125}$$

for the sake of algorithmic simplicity, yielding

$$\widehat{p}_{ml}(t) \approx \widehat{z}_{ml}^{(*,*)}(t) - \widehat{s}_{ml}(t-1)\underbrace{\left(\sum_{i=1}^{N_b} \widehat{z}_{ml}^{(i,*)}(t)^2\nu_i^b(t) + \sum_{j=1}^{N_c} \widehat{z}_{ml}^{(*,j)}(t)^2\nu_j^c(t)\right)}_{\triangleq\, \overline{\nu}_{ml}^{p}(t)}, \tag{5.126}$$

and note that similar approximations were made for BiG-AMP in [8], where empirical tests showed little effect. Of course, a more complicated version of the P-BiG-AMP algorithm could be stated using (5.123) instead of (5.126).

Equations (5.119)-(5.120) can also be used to simplify $\nu_{ml}^p(t)$. For this, we first use the facts $\nu_{ml,j}^c(t) = \nu_j^c(t) + O(1/N^3)$ and $\nu_{ml,i}^b(t) = \nu_i^b(t) + O(1/N^3)$ to write (5.54) as

$$\nu_{ml}^p(t) = \sum_{j=1}^{N_c} \nu_j^c(t)\widehat{z}_{\to ml}^{(*,j)}(t)^2 + \sum_{i=1}^{N_b} \nu_i^b(t)\widehat{z}_{\to ml}^{(i,*)}(t)^2 + \sum_{i=1}^{N_b}\sum_{j=1}^{N_c} \nu_i^b(t)\nu_j^c(t)\widehat{z}_{ml}^{(i,j)2} + O(1/N). \tag{5.127}$$

Then we use (5.119) to write (5.42) as

$$\widehat{z}_{\to ml}^{(i,*)}(t) = \sum_{n=1}^{N} a_{mn}^{(i)}\left(x_{nl}\big(\widehat{\boldsymbol{c}}(t)\big) - \widehat{s}_{ml}(t-1)\sum_{j=1}^{N_c} \widehat{z}_{ml}^{(*,j)}(t-1)\nu_j^c(t)x_{nl}^{(j)}\right) + O(1/N^2) \tag{5.128}$$

$$= \widehat{z}_{ml}^{(i,*)}(t) - \widehat{s}_{ml}(t-1)\sum_{j=1}^{N_c} \widehat{z}_{ml}^{(*,j)}(t-1)\widehat{z}_{ml}^{(i,j)}\nu_j^c(t) + O(1/N^2), \tag{5.129}$$

and (5.120) to write (5.43) as

$$\widehat{z}^{(*,j)}_{\to ml}(t) = \sum_{n=1}^{N} x^{(j)}_{nl} \left( a_{mn}\big(\widehat{\boldsymbol{b}}(t)\big) - \widehat{s}_{ml}(t-1) \sum_{i=1}^{N_b} \widehat{z}^{(i,*)}_{ml}(t-1) \nu^b_i(t) a^{(i)}_{mn} \right) + O(1/N^2) \tag{5.130}$$

$$= \widehat{z}^{(*,j)}_{ml}(t) - \widehat{s}_{ml}(t-1) \sum_{i=1}^{N_b} \widehat{z}^{(i,*)}_{ml}(t-1) \widehat{z}^{(i,j)}_{ml} \nu^b_i(t) + O(1/N^2), \tag{5.131}$$

which can be plugged into (5.127) to yield

$$\begin{aligned}
\nu^p_{ml}(t) = & \sum_{j=1}^{N_c} \nu^c_j(t) \left( \widehat{z}^{(*,j)}_{ml}(t) - \widehat{s}_{ml}(t-1) \sum_{i=1}^{N_b} \widehat{z}^{(i,*)}_{ml}(t-1) \widehat{z}^{(i,j)}_{ml} \nu^b_i(t) \right)^2 \\
& + \sum_{i=1}^{N_b} \nu^b_i(t) \left( \widehat{z}^{(i,*)}_{ml}(t) - \widehat{s}_{ml}(t-1) \sum_{j=1}^{N_c} \widehat{z}^{(*,j)}_{ml}(t-1) \widehat{z}^{(i,j)}_{ml} \nu^c_j(t) \right)^2 \\
& + \sum_{i=1}^{N_b} \sum_{j=1}^{N_c} \nu^b_i(t) \nu^c_j(t) \widehat{z}^{(i,j)2}_{ml} + O(1/N) \qquad (5.132) \\
= & \, \overline{\nu}^p_{ml}(t) + \sum_{i=1}^{N_b} \sum_{j=1}^{N_c} \nu^b_i(t) \nu^c_j(t) \widehat{z}^{(i,j)2}_{ml} \\
& - 2\widehat{s}_{ml}(t-1) \left[ \sum_{j=1}^{N_c} \nu^c_j(t) \widehat{z}^{(*,j)}_{ml}(t) \sum_{i=1}^{N_b} \widehat{z}^{(i,*)}_{ml}(t-1) \widehat{z}^{(i,j)}_{ml} \nu^b_i(t) \right. \\
& \quad \left. + \sum_{i=1}^{N_b} \nu^b_i(t) \widehat{z}^{(i,*)}_{ml}(t) \sum_{j=1}^{N_c} \widehat{z}^{(*,j)}_{ml}(t-1) \widehat{z}^{(i,j)}_{ml} \nu^c_j(t) \right] \\
& + \widehat{s}^2_{ml}(t-1) \left[ \sum_{j=1}^{N_c} \nu^c_j(t) \left( \sum_{i=1}^{N_b} \widehat{z}^{(i,*)}_{ml}(t-1) \widehat{z}^{(i,j)}_{ml} \nu^b_i(t) \right)^2 \right. \\
& \quad \left. + \sum_{i=1}^{N_b} \nu^b_i(t) \left( \sum_{j=1}^{N_c} \widehat{z}^{(*,j)}_{ml}(t-1) \widehat{z}^{(i,j)}_{ml} \nu^c_j(t) \right)^2 \right] \\
& + O(1/N) \qquad (5.133) \\
\approx & \, \overline{\nu}^p_{ml}(t) + \sum_{i=1}^{N_b} \sum_{j=1}^{N_c} \nu^b_i(t) \nu^c_j(t) \widehat{z}^{(i,j)2}_{ml}, \qquad (5.134)
\end{aligned}$$

where we have again retained only the $O(1)$ terms, since the others vanish in the large-system limit.

Next, we eliminate the dependence on $\widehat{z}^{(*,j)}_{\rightarrow ml}(t)$ from $\widehat{r}_j(t)$. Plugging (5.131) into (5.98), we obtain

$$\widehat{r}_j(t) = \widehat{c}_j(t) + \nu^r_j(t) \sum_{m,l} \left(\widehat{s}^2_{ml}(t) - \nu^s_{ml}(t)\right) \sum_{i=1}^{N_b} \nu^b_i(t) \widehat{z}^{(i,j)}_{ml} \widehat{z}^{(i,*)}_{ml}(t)$$
$$+ \nu^r_j(t) \sum_{m,l} \widehat{s}_{ml}(t) \left( \widehat{z}^{(*,j)}_{ml}(t) - \widehat{s}_{ml}(t-1) \sum_{i=1}^{N_b} \widehat{z}^{(i,*)}_{ml}(t-1) \widehat{z}^{(i,j)}_{ml} \nu^b_i(t) \right) + O(1/N^3) \tag{5.135}$$

$$\approx \widehat{c}_j(t) + \nu^r_j(t) \sum_{m,l} \left(\widehat{s}^2_{ml}(t) - \nu^s_{ml}(t)\right) \sum_{i=1}^{N_b} \nu^b_i(t) \widehat{z}^{(i,j)}_{ml} \widehat{z}^{(i,*)}_{ml}(t)$$
$$+ \nu^r_j(t) \sum_{m,l} \widehat{s}_{ml}(t) \left( \widehat{z}^{(*,j)}_{ml}(t) - \widehat{s}_{ml}(t-1) \sum_{i=1}^{N_b} \widehat{z}^{(i,*)}_{ml}(t-1) \widehat{z}^{(i,j)}_{ml} \nu^b_i(t) \right), \tag{5.136}$$

neglecting terms that vanish in the large-system limit. We note that the order of the neglected term, $O(1/N^3)$, is consistent with the approximation term in (5.102). Although not justified by the large-system limit, we will also employ the approximation

$$\sum_{m,l} \widehat{s}^2_{ml}(t) \sum_{i=1}^{N_b} \nu^b_i(t) \widehat{z}^{(i,j)}_{ml} \widehat{z}^{(i,*)}_{ml}(t) \approx \sum_{m,l} \widehat{s}_{ml}(t) \widehat{s}_{ml}(t-1) \sum_{i=1}^{N_b} \nu^b_i(t) \widehat{z}^{(i,j)}_{ml} \widehat{z}^{(i,*)}_{ml}(t-1) \tag{5.137}$$

for the sake of algorithmic simplicity, yielding

$$\widehat{r}_j(t) \approx \widehat{c}_j(t) + \nu^r_j(t) \sum_{m,l} \left( \widehat{s}_{ml}(t) \widehat{z}^{(*,j)}_{ml}(t) - \nu^s_{ml}(t) \sum_{i=1}^{N_b} \nu^b_i(t) \widehat{z}^{(i,j)}_{ml} \widehat{z}^{(i,*)}_{ml}(t) \right). \tag{5.138}$$

noting that similar approximations were also made for BiG-AMP [8]. Similarly, we substitute (5.129) into (5.112) and make analogous approximations to obtain

$$\widehat{q}_i(t) \approx \widehat{b}_i(t) + \nu^q_i(t) \sum_{m,l} \left( \widehat{s}_{ml}(t) \widehat{z}^{(i,*)}_{ml}(t) - \nu^s_{ml}(t) \sum_{j=1}^{N_c} \nu^c_j(t) \widehat{z}^{(i,j)}_{ml} \widehat{z}^{(*,j)}_{ml}(t) \right). \tag{5.139}$$

Next, we simply expressions for the variances $\nu_j^r(t)$ and $\nu_i^q(t)$. For this, we first note that the second half of $\nu_j^r(t)$ from (5.97) can be written as

$$\sum_{m,l}\big(\widehat{s}_{ml}^2(t)-\nu_{ml}^s(t)\big)\sum_{i=1}^{N_b}\nu_i^b(t)\widehat{z}_{ml}^{(i,j)2} \tag{5.140}$$

$$=\sum_{m,l}\left[\left(\frac{\widehat{z}_{ml}(t)-\widehat{p}_{ml}(t)}{\nu_{ml}^p(t)}\right)^2-\frac{1}{\nu_{ml}^p(t)}\left(1-\frac{\nu_{ml}^z(t)}{\nu_{ml}^p(t)}\right)\right]\sum_{i=1}^{N_b}\nu_i^b(t)\widehat{z}_{ml}^{(i,j)2} \tag{5.141}$$

$$=\sum_{m,l}\left(\frac{\big(\widehat{z}_{ml}(t)-\widehat{p}_{ml}(t)\big)^2+\nu_{ml}^z(t)}{\nu_{ml}^p(t)}-1\right)\frac{\sum_{i=1}^{N_b}\nu_i^b(t)\widehat{z}_{ml}^{(i,j)2}}{\nu_{ml}^p(t)} \tag{5.142}$$

$$=\sum_{m,l}\left(\mathrm{E}\left\{\frac{\big(\mathsf{z}_{ml}-\widehat{p}_{ml}(t)\big)^2}{\nu_{ml}^p(t)}\right\}-1\right)\frac{\sum_{i=1}^{N_b}\nu_i^b(t)\widehat{z}_{ml}^{(i,j)2}}{\nu_{ml}^p(t)}, \tag{5.143}$$

where the expressions for $\widehat{s}_{ml}^2(t)$ and $\nu_{ml}^s(t)$ came from (5.74) and (5.75) and the random variable $\mathsf{z}_{ml}$ above is distributed according to the pdf in (5.78). For the simpler GAMP algorithm, [6, Sec. VI.D] clarifies that, under i.i.d priors and scalar variances, in the large-system limit, the true $z_m$ and the GAMP iterates $\widehat{p}_m(t)$ converge empirically to a pair of random variables $(\mathsf{z},\mathsf{p})$ that satisfy $p_{\mathsf{z}|\mathsf{p}}(z|\widehat{p}(t))=\mathcal{N}(z;\widehat{p}(t),\nu^p(t))$. This leads us to believe that (5.143) is negligible in the large-system limit, yielding the approximation

$$\nu_j^r(t)\approx\left(\sum_{m,l}\nu_{ml}^s(t)\widehat{z}_{ml}^{(*,j)}(t)^2\right)^{-1}. \tag{5.144}$$

A similar argument yields

$$\nu_i^q(t)\approx\left(\sum_{m,l}\nu_{ml}^s(t)\widehat{z}_{ml}^{(i,*)}(t)^2\right)^{-1}. \tag{5.145}$$

The final step in our P-BiG-AMP derivation approximates the SPA posterior log-pdfs in (5.27) and (5.28). Plugging (5.70) and (5.89) into these expressions, we get

$$\Delta_i^{\mathsf{b}}(t+1,b_i)\approx\mathsf{const}+\log\Big(p_{\mathsf{b}_i}(b_i)\mathcal{N}\big(b_i;\widehat{q}_i(t),\nu_i^q(t)\big)\Big) \tag{5.146}$$

$$\Delta_j^{\mathsf{c}}(t+1,c_j)\approx\mathsf{const}+\log\Big(p_{\mathsf{c}_j}(c_j)\mathcal{N}\big(c_j;\widehat{r}_j(t),\nu_j^r(t)\big)\Big), \tag{5.147}$$

using steps similar to those used for (5.92). The associated pdfs are given as (D2) and (D3) in Table 5.3 and represent P-BiG-AMP's iteration-$t$ approximations to the true marginal posteriors $p_{\mathsf{b}_i|\mathsf{Y}}(b_i \,|\, \boldsymbol{Y})$ and $p_{\mathsf{c}_j|\mathsf{Y}}(c_j \,|\, \boldsymbol{Y})$. The quantities $\widehat{b}_i(t{+}1)$ and $\nu^b_i(t{+}1)$ are then defined as the mean and variance, respectiveley, of the pdf associated with (5.146), and $\widehat{c}_j(t{+}1)$ and $\nu^c_j(t{+}1)$ are the mean and variancce of the pdf associated with (5.147). As such, $\widehat{b}_i(t{+}1)$ represents P-BiG-AMP's approximation to the MMSE estimate of $\mathsf{b}_i$ and $\nu^b_i(t{+}1)$ represents its approximation of the corresponding MSE. Likewise, $\widehat{c}_j(t{+}1)$ represents P-BiG-AMP's approximation to the MMSE estimate of $\mathsf{c}_j$ and $\nu^c_j(t{+}1)$ represents its approximation of the corresponding MSE. This completes the derivation of P-BiG-AMP.

### 5.5.6 Algorithm Summary

The P-BiG-AMP algorithm is summarized in Table 5.3. The version stated in the table includes a maximum number of iterations $T_{\max}$, as well as a stopping condition (R24) that terminates the iterations when the change in the residual $\widehat{z}^{(*,*)}_{ml}(t)$ falls below a user-defined parameter $\tau_{\text{stop}}$. Noting the complex conjugates in (R17) and (R19), the stated version also allows the use of *complex-valued* quantities, in which case $\mathcal{N}$ in (D1)-(D3) would denote a circular complex Gaussian pdf. However, for ease of interpretation, Table 5.3 does not include the important damping modifications that will be detailed in Section 5.8.

Furthermore, Table 5.3 is written in a way that facilitates the use of *non-linear* parameterizations $\boldsymbol{A}(\cdot)$ and $\boldsymbol{X}(\cdot)$, where the corresponding element-wise partials $a^{(i)}_{mn}\big(\boldsymbol{b}\big)$ and $x^{(j)}_{nl}\big(\boldsymbol{c}\big)$ can vary with $\boldsymbol{b}$ and $\boldsymbol{c}$, as in the Taylor-series expansion (5.4)-(5.5). This is visible from steps (R3)-(R4), as well as from the $t$-dependence on $\widehat{z}^{(i,j)}_{ml}(t)$. In the case

definitions:

$$p_{\mathsf{z}_{ml}|\mathsf{p}_{ml}}\big(z\,|\,\widehat{p};\nu^p\big) \triangleq \frac{p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml}\,|\,z)\,\mathcal{N}\big(z;\widehat{p},\nu^p\big)}{\int_{z'} p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml}\,|\,z')\,\mathcal{N}\big(z';\widehat{p},\nu^p\big)} \tag{D1}$$

$$p_{\mathsf{c}_j|\mathsf{r}_j}\big(c\,|\,\widehat{r};\nu^r\big) \triangleq \frac{p_{\mathsf{c}_j}(c)\,\mathcal{N}\big(c;\widehat{r},\nu^r\big)}{\int_{c'} p_{\mathsf{c}_j}(c')\,\mathcal{N}\big(c';\widehat{r},\nu^r\big)} \tag{D2}$$

$$p_{\mathsf{b}_i|\mathsf{q}_i}\big(b\,|\,\widehat{q};\nu^q\big) \triangleq \frac{p_{\mathsf{b}_i}(b)\,\mathcal{N}\big(b;\widehat{q},\nu^q\big)}{\int_{b'} p_{\mathsf{b}_i}(b')\,\mathcal{N}\big(b';\widehat{q},\nu^q\big)} \tag{D3}$$

initialization:

$$\forall m,l: \widehat{s}_{ml}(0) = 0 \tag{I1}$$

$$\forall i,j: \text{choose} \quad \widehat{b}_i(1), \nu^b_i(1), \widehat{c}_j(1), \nu^c_j(1) \tag{I2}$$

for $t = 1, \ldots T_{\max}$

$$\forall m,n: \widehat{a}_{mn}(t) = a_{mn}\big(\widehat{\boldsymbol{b}}(t)\big) \tag{R1}$$

$$\forall n,l: \widehat{x}_{nl}(t) = x_{nl}\big(\widehat{\boldsymbol{c}}(t)\big) \tag{R2}$$

$$\forall m,n,i: \widehat{a}^{(i)}_{mn}(t) = a^{(i)}_{mn}\big(\widehat{\boldsymbol{b}}(t)\big) \tag{R3}$$

$$\forall n,l,j: \widehat{x}^{(j)}_{nl}(t) = x^{(j)}_{nl}\big(\widehat{\boldsymbol{c}}(t)\big) \tag{R4}$$

$$\forall m,l,i,j: \widehat{z}^{(i,j)}_{ml}(t) = \textstyle\sum_{n=1}^N \widehat{a}^{(i)}_{mn}(t)\widehat{x}^{(j)}_{nl}(t) \tag{R5}$$

$$\forall m,l,i: \widehat{z}^{(i,*)}_{ml}(t) = \textstyle\sum_{n=1}^N \widehat{a}^{(i)}_{mn}(t)\widehat{x}_{nl}(t) \tag{R6}$$

$$\forall m,l,j: \widehat{z}^{(*,j)}_{ml}(t) = \textstyle\sum_{n=1}^N \widehat{a}_{mn}(t)\widehat{x}^{(j)}_{nl}(t) \tag{R7}$$

$$\forall m,l: \widehat{z}^{(*,*)}_{ml}(t) = \textstyle\sum_{n=1}^N \widehat{a}_{mn}(t)\widehat{x}_{nl}(t) \tag{R8}$$

$$\forall m,l: \overline{\nu}^p_{ml}(t) = \textstyle\sum_{i=1}^{N_b} |\widehat{z}^{(i,*)}_{ml}(t)|^2\nu^b_i(t) + \sum_{j=1}^{N_c} |\widehat{z}^{(*,j)}_{ml}(t)|^2\nu^c_j(t) \tag{R9}$$

$$\forall m,l: \nu^p_{ml}(t) = \overline{\nu}^p_{ml}(t) + \textstyle\sum_{i=1}^{N_b}\sum_{j=1}^{N_c} \nu^b_i(t)\nu^c_j(t)|\widehat{z}^{(i,j)}_{ml}(t)|^2 \tag{R10}$$

$$\forall m,l: \widehat{p}_{ml}(t) = \widehat{z}^{(*,*)}_{ml}(t) - \widehat{s}_{ml}(t-1)\overline{\nu}^p_{ml}(t) \tag{R11}$$

$$\forall m,l: \nu^z_{ml}(t) = \mathrm{var}\{\mathsf{z}_{ml}\,|\,\mathsf{p}_{ml}=\widehat{p}_{ml}(t);\nu^p_{ml}(t)\} \tag{R12}$$

$$\forall m,l: \widehat{z}_{ml}(t) = \mathrm{E}\{\mathsf{z}_{ml}\,|\,\mathsf{p}_{ml}=\widehat{p}_{ml}(t);\nu^p_{ml}(t)\} \tag{R13}$$

$$\forall m,l: \nu^s_{ml}(t) = (1-\nu^z_{ml}(t)/\nu^p_{ml}(t))/\nu^p_{ml}(t) \tag{R14}$$

$$\forall m,l: \widehat{s}_{ml}(t) = (\widehat{z}_{ml}(t) - \widehat{p}_{ml}(t))/\nu^p_{ml}(t) \tag{R15}$$

$$\forall j: \nu^r_j(t) = \Big(\textstyle\sum_{m,l} \nu^s_{ml}(t)|\widehat{z}^{(*,j)}_{ml}(t)|^2\Big)^{-1} \tag{R16}$$

$$\forall j: \widehat{r}_j(t) = \widehat{c}_j(t) + \nu^r_j(t)\textstyle\sum_{m,l}\Big(\widehat{s}_{ml}(t)\widehat{z}^{(*,j)}_{ml}(t)^* - \nu^s_{ml}(t)\sum_{i=1}^{N_b}\nu^b_i(t)\widehat{z}^{(i,j)}_{ml}(t)^*\widehat{z}^{(i,*)}_{ml}(t)\Big) \tag{R17}$$

$$\forall i: \nu^q_i(t) = \Big(\textstyle\sum_{m,l} \nu^s_{ml}(t)|\widehat{z}^{(i,*)}_{ml}(t)|^2\Big)^{-1} \tag{R18}$$

$$\forall i: \quad \widehat{q}_i(t) = \widehat{b}_i(t) + \nu^q_i(t)\textstyle\sum_{m,l}\Big(\widehat{s}_{ml}(t)\widehat{z}^{(i,*)}_{ml}(t)^* - \nu^s_{ml}(t)\sum_{j=1}^{N_c}\nu^c_j(t)\widehat{z}^{(i,j)}_{ml}(t)^*\widehat{z}^{(*,j)}_{ml}(t)\Big) \tag{R19}$$

$$\forall j: \quad \nu^c_j(t+1) = \mathrm{var}\{\mathsf{c}_j\,|\,\mathsf{r}_j=\widehat{r}_j(t);\nu^r_j(t)\} \tag{R20}$$

$$\forall j: \quad \widehat{c}_j(t+1) = \mathrm{E}\{\mathsf{c}_j\,|\,\mathsf{r}_j=\widehat{r}_j(t);\nu^r_j(t)\} \tag{R21}$$

$$\forall i: \quad \nu^b_i(t+1) = \mathrm{var}\{\mathsf{b}_i\,|\,\mathsf{q}_i=\widehat{q}_i(t);\nu^q_i(t)\} \tag{R22}$$

$$\forall i: \quad \widehat{b}_i(t+1) = \mathrm{E}\{\mathsf{b}_i\,|\,\mathsf{q}_i=\widehat{q}_i(t);\nu^q_i(t)\} \tag{R23}$$

$$\text{if } \textstyle\sum_{m,l}|\widehat{z}^{(*,*)}_{ml}(t) - \widehat{z}^{(*,*)}_{ml}(t-1)|^2 \le \tau_{\text{stop}}\sum_{m,l}|\widehat{z}^{(*,*)}_{ml}(t)|^2, \text{ stop} \tag{R24}$$

end

Table 5.3: The P-BiG-AMP Algorithm

that $\boldsymbol{A}(\cdot)$ and/or $\boldsymbol{X}(\cdot)$ are nonlinear, steps (R1)-(R4) show that P-BiG-AMP takes a first-order Taylor series approximation of $\boldsymbol{A}(\cdot)$ and $\boldsymbol{X}(\cdot)$ about the current-iteration's parameter estimates $\widehat{\boldsymbol{b}}(t)$ and $\widehat{\boldsymbol{c}}(t)$, similar to the Extended Kalman Filter [88]. By contrast, in the simpler *affine* case assumed throughout the derivation, the partial derivatives $a_{mn}^{(i)}\big(\boldsymbol{b}\big)$ and $x_{nl}^{(j)}\big(\boldsymbol{c}\big)$ are invariant to $\boldsymbol{b}$ and $\boldsymbol{c}$, as in (5.8)-(5.9), in which case steps (R3)-(R4) become trivial.

The initializations used in step (I2) are application specific. In many cases, random draws from the prior distributions, along with large initial variances, is a reasonable choice. With sparse coefficients, initializing either $\widehat{\boldsymbol{b}}(1)$ or $\widehat{\boldsymbol{c}}(1)$ with zeros can also be effective.

The number of multiplies required by each step of Table 5.3 is tabulated in Table 5.4. The complexity of steps (R1)-(R4) cannot be characterized in generality because it depends strongly on the nature of the parameterizations $\boldsymbol{A}(\cdot)$ and $\boldsymbol{X}(\cdot)$. Similarly, the number of multiplies required in steps (R12)-(R13) and (R20)-(R23) depend strongly on the prior and the likelihood, respectively. Thus, for (R1), we denote the complexity as $f_{\text{R1}}(N_b, N, M)$, which is some function of the dimensions $N_b$, $N$, and $M$, and we similar for steps (R2)-(R4), (R12)-(R13), and (R20)-(R23). In fact, as we shall see in the sequel, certain classes of $\boldsymbol{A}(\cdot)$ and $\boldsymbol{X}(\cdot)$ allow simplifications to be made to other steps in Table 5.3, and thus the complexities reported in Table 5.4 should be interpreted as "worst case" values.

| | | | | | |
|---|---|---|---|---|---|
| (R1) | $f_{\text{R1}}(N_b, M, N)$ | (R2) | $f_{\text{R2}}(N_c, N, L)$ | (R3) | $f_{\text{R3}}(N_b, M, N)$ |
| (R4) | $f_{\text{R4}}(N_c, N, L)$ | (R5) | $N_b N_c MNL$ | (R6) | $N_b MNL$ |
| (R7) | $N_c MNL$ | (R8) | $NML$ | (R9) | $2(N_b + N_c)ML$ |
| (R10) | $3N_b N_c ML$ | (R11) | $ML$ | (R12) | $f_{\text{R12}}(M, L)$ |
| (R13) | $f_{\text{R13}}(M, L)$ | (R14) | $2ML$ | (R15) | $ML$ |
| (R16) | $2N_c ML$ | (R17) | $2N_b N_c ML$ $+N_c ML$ | (R18) | $2N_b N_c ML$ $+N_b ML$ |
| (R19) | $2N_b ML$ | (R20) | $f_{\text{R20}}(N_c)$ | (R21) | $f_{\text{R21}}(N_c)$ |
| (R22) | $f_{\text{R22}}(N_b)$ | (R23) | $f_{\text{R23}}(N_b)$ | | |

Table 5.4: Multiplies consumed by each step of P-BiG-AMP from Table 5.3.

## 5.6 Special Parameterizations

We now describe several special cases of the parameterizations $\boldsymbol{A}(\cdot)$ and $\boldsymbol{X}(\cdot)$ that arise commonly in practice and the corresponding simplifications that can be made to P-BiG-AMP.

### 5.6.1 Affine Parameterizations

First we consider the case where both $\boldsymbol{A}(\cdot)$ and $\boldsymbol{X}(\cdot)$ are affine, as defined in (5.6)-(5.7) and assumed throughout the derivation of P-BiG-AMP. In this case, the partial derivatives $\widehat{a}_{mn}^{(i)}(t) = a_{mn}^{(i)}\big(\widehat{\boldsymbol{b}}(t)\big)$ and $\widehat{x}_{nl}^{(j)}(t) = x_{nl}^{(j)}\big(\widehat{\boldsymbol{c}}(t)\big)$ in steps (R3) and (R4) of Table 5.3 are invariant to $\widehat{\boldsymbol{b}}(t)$ and $\widehat{\boldsymbol{c}}(t)$, and thus can be pre-computed, relieving the algorithm of further calls to (R3)-(R4). Consequently, $\widehat{z}_{ml}^{(i,j)}(t)$ from step (R5) can also be pre-computed and used for all $t$, relieving the algorithm of further calls to (R5).

Furthermore, the pre-computation of $\widehat{z}_{ml}^{(i,j)}(t)$ facilitates an alternative approach to steps (R1)-(R2) and (R6)-(R8) that bypasses the explicit evaluation of $\boldsymbol{A}(\cdot)$ and

$\boldsymbol{X}(\cdot)$. In particular, using (5.6)-(5.7), steps (R2) and (R6) reduce to

$$\widehat{z}_{ml}^{(i,*)}(t) = \sum_{n=1}^{N} a_{mn}^{(i)} x_{nl}\big(\widehat{\boldsymbol{c}}(t)\big) = \sum_{n=1}^{N} a_{mn}^{(i)} \sum_{j=0}^{N_c} \widehat{c}_j(t) x_{nl}^{(j)} = \sum_{j=0}^{N_c} \widehat{c}_j(t) \sum_{n=1}^{N} a_{mn}^{(i)} x_{nl}^{(j)} \tag{5.148}$$

$$= \sum_{j=0}^{N_c} \widehat{c}_j(t) \widehat{z}_{ml}^{(i,j)}. \tag{5.149}$$

In a similar manner, steps (R1) and (R7) reduce to

$$\widehat{z}_{ml}^{(*,j)}(t) = \sum_{i=0}^{N_b} \widehat{b}_i(t) \widehat{z}_{ml}^{(i,j)} \tag{5.150}$$

and step (R8) reduces to

$$\widehat{z}_{ml}^{(*,*)}(t) = \sum_{n=1}^{N} a_{mn}\big(\widehat{\boldsymbol{b}}(t)\big) x_{nl}\big(\widehat{\boldsymbol{c}}(t)\big) = \sum_{n=1}^{N} \sum_{i=0}^{N_b} \widehat{b}_i(t) a_{mn}^{(i)} \sum_{j=0}^{N_c} \widehat{c}_j(t) x_{nl}^{(j)} \tag{5.151}$$

$$= \sum_{i=0}^{N_b} \sum_{j=0}^{N_c} \widehat{b}_i(t) \widehat{c}_j(t) \widehat{z}_{ml}^{(i,j)} \tag{5.152}$$

$$= \sum_{i=0}^{N_b} \widehat{b}_i(t) \widehat{z}_{ml}^{(i,*)}(t) = \sum_{j=0}^{N_c} \widehat{c}_j(t) \widehat{z}_{ml}^{(*,j)}(t). \tag{5.153}$$

We note that, in the expressions above, $\widehat{b}_0(t) = 1/\sqrt{N_b}\ \forall t$ and $\widehat{c}_0(t) = 1/\sqrt{N_c}\ \forall t$, because, in (5.6)-(5.7), the true values of these estimates, $b_0 = 1/\sqrt{N_b}$ and $c_0 = 1/\sqrt{N_c}$, are known constants.

A natural question arises as to whether it is computationally advantageous to use (5.149), (5.150), and (5.153) in place of (R1)-(R2) and (R6)-(R8). To answer this question, we first note that evaluating (5.149) for all $m, l, i$ consumes $N_b N_c ML$ multiplies,[45] evaluating (5.150) for all $m, l, j$ consumes another $N_b N_c ML$ multiplies, and evaluating (5.153) for all $m, l$ consumes $\min(N_b, N_c)ML$ multiplies, for a grand total of $\approx 2N_b N_c ML$ multiplies. Meanwhile, direct computation of (R1) and (R2) via

[45]To simplify the exposition, we neglect the extra multiply associated with the $i, j = 0$ terms in the sums.

(5.6) and (5.7) consumes $N_bMN$ and $N_cNL$ multiplies, respectively, and Table 5.4 specifies that (R6)-(R8) consume a total of $(N_b + N_c + 1)NML$ multiplies, for a grand total of $\approx (N_bN + N_cN)ML$ multiplies. Thus, when $N_b \ll N$ and $N_c \ll N$, the computational savings from (5.149), (5.150), and (5.153) can be substantial. Conversely, when $N_b \gg N$ or $N_c \gg N$, the use of (5.149), (5.150), and (5.153) can be counterproductive. Later, we discuss the special case where $\boldsymbol{A}(\cdot)$ and/or $\boldsymbol{X}(\cdot)$ have a fast implementation (e.g., FFT) that can be used to circumvent direct evaluation of (R1) and/or (R2), in which case the computational tradeoff must be re-evaluated.

### 5.6.2 Trivial Parameterizations

Next we consider the case of trivial parameterizations, where the elements in $\boldsymbol{A}$ (or $\boldsymbol{X}$) can be put in one-to-one correspondence with the elements in $\boldsymbol{b}$ (or $\boldsymbol{c}$). Concretely, we refer to $\boldsymbol{A}(\cdot)$ as a "trivial parameterization" when $[\boldsymbol{A}(\boldsymbol{b})]_{mn} = [\boldsymbol{b}]_{\phi_{\mathsf{b}}(m,n)}$ for some one-to-one index map $\phi_{\mathsf{b}}(\cdot,\cdot) : \{1...M\} \times \{1...N\} \to \{1...N_b\}$, which requires that $N_b = MN$. Similarly, $\boldsymbol{X}(\cdot)$ is a trivial parameterization when $[\boldsymbol{X}(\boldsymbol{c})]_{nl} = [\boldsymbol{c}]_{\phi_{\mathsf{c}}(n,l)}$ for some one-to-one index map $\phi_{\mathsf{c}}(\cdot,\cdot) : \{1...N\} \times \{1...L\} \to \{1...N_c\}$, which requires that $N_c = NL$. As an example, the case of column-major vectorizations $\boldsymbol{b} = \mathrm{vec}(\boldsymbol{A})$ and $\boldsymbol{c} = \mathrm{vec}(\boldsymbol{X})$ yields the index maps $\phi_{\mathsf{b}}(m,n) = m+(n-1)M$ and $\phi_{\mathsf{c}}(n,l) = n+(l-1)N$. For use in the sequel, we note that trivial parameterizations have partial derivatives of the form

$$a_{mn}^{(i)}(\boldsymbol{b}) = \begin{cases} 1 & i = \phi_{\mathsf{b}}(m,n) \\ 0 & i \neq \phi_{\mathsf{b}}(m,n) \end{cases} \quad \forall \boldsymbol{b} \tag{5.154}$$

$$x_{nl}^{(j)}(\boldsymbol{c}) = \begin{cases} 1 & j = \phi_{\mathsf{c}}(n,l) \\ 0 & j \neq \phi_{\mathsf{c}}(n,l) \end{cases} \quad \forall \boldsymbol{c}. \tag{5.155}$$

**One Parameterization is Trivial**

We now show that P-BiG-AMP simplifies even when only one of the parameterizations $\boldsymbol{A}(\cdot)$ or $\boldsymbol{X}(\cdot)$ is trivial. For the sake of brevity, we only consider the case where $\boldsymbol{X}(\cdot)$ is trivial, noting that a similar treatment can be applied ot the case where $\boldsymbol{A}(\cdot)$ is trivial.

First we recall that, by definition, trivial $\boldsymbol{X}(\cdot)$ implies

$$\widehat{x}_{nl}(t) = x_{nl}\big(\widehat{\boldsymbol{c}}(t)\big) = \widehat{c}_{\phi_{\mathsf{c}}(n,l)}(t). \tag{5.156}$$

Thus, steps (R2), (R6), and (R8) in Table 5.3 reduce to

$$\widehat{z}^{(i,*)}_{ml}(t) = \sum_{n=1}^{N} \widehat{a}^{(i)}_{mn}(t)\widehat{x}_{nl}(t) = \sum_{n=1}^{N} \widehat{a}^{(i)}_{mn}(t)\widehat{c}_{\phi_{\mathsf{c}}(n,l)}(t) \tag{5.157}$$

$$\widehat{z}^{(*,*)}_{ml}(t) = \sum_{n=1}^{N} \widehat{a}_{mn}(t)\widehat{x}_{nl}(t) = \sum_{n=1}^{N} \widehat{a}_{mn}(t)\widehat{c}_{\phi_{\mathsf{c}}(n,l)}(t), \tag{5.158}$$

which can be implemented efficiently, via matrix multiplication, after appropriately reshaping $\widehat{\boldsymbol{c}}(t)$.

Next we recall that (5.155) holds for trivial $\boldsymbol{X}(\cdot)$. There, for each fixed pair $(j,l)$, there exists at most one (but possibly no) value of $n \in \{1...N\}$ that satisfies $j = \phi_{\mathsf{c}}(n,l)$. In other words, given an index $j$ into $\boldsymbol{b}$ and a column index $l$ in $\boldsymbol{X}$, there exists at most one row index $n$ such that $[\boldsymbol{X}]_{n,l} = b_j$. We now define a mapping $\varphi_{\mathsf{c}}(\cdot,\cdot) : \{0...N_c\} \times \{1...L\} \to \{0...N\}$ such that $\varphi_{\mathsf{c}}(j,l) = n \in \{1...N\}$ if such a row index $n$ exists and $\varphi_{\mathsf{c}}(j,l) = 0$ if no such $n$ exists. Then, similar to (5.155), a trivial $\boldsymbol{X}(\cdot)$ ensures that, for any $n \in \{1...N\}$, $l \in \{1...L\}$, $j \in \{0...N_c\}$:

$$\widehat{x}^{(j)}_{nl}(t) = x^{(j)}_{nl}\big(\widehat{\boldsymbol{c}}(t)\big) = \begin{cases} 1 & n = \varphi_{\mathsf{c}}(j,l) \\ 0 & n \neq \varphi_{\mathsf{c}}(j,l) \end{cases}. \tag{5.159}$$

Thus, steps (R4), (R5), and (R7) reduce to

$$\widehat{z}_{ml}^{(i,j)}(t) = \sum_{n=1}^{N} \widehat{a}_{mn}^{(i)}(t)\widehat{x}_{nl}^{(j)}(t) = \begin{cases} \widehat{a}_{m,\varphi_{\mathsf{c}}(j,l)}^{(i)}(t) & \varphi_{\mathsf{c}}(j,l) \in \{1...N\} \\ 0 & \varphi_{\mathsf{c}}(j,l) = 0 \end{cases} \tag{5.160}$$

$$\widehat{z}_{ml}^{(*,j)}(t) = \sum_{n=1}^{N} \widehat{a}_{mn}(t)\widehat{x}_{nl}^{(j)}(t) = \begin{cases} \widehat{a}_{m,\varphi_{\mathsf{c}}(j,l)}(t) & \varphi_{\mathsf{c}}(j,l) \in \{1...N\} \\ 0 & \varphi_{\mathsf{c}}(j,l) = 0, \end{cases} \tag{5.161}$$

which require no computation and allow subsequent steps based on $\widehat{z}_{ml}^{(*,j)}(t)$ to be computed using only elements from $\widehat{\boldsymbol{A}}(t)$.

Many other steps[46] in the algorithm can also be computed efficiently by operating on matrix-shaped versions of $\widehat{\boldsymbol{c}}(t)$, $\{\nu_j^c\}$, $\widehat{\boldsymbol{r}}$, and $\{\nu_j^r\}$, with similarities to the corresponding steps in BiG-AMP.

**Both Parameterizations are Trivial**

We now show that, when both $\boldsymbol{A}(\cdot)$ and $\boldsymbol{X}(\cdot)$ are trivial, the P-BiG-AMP algorithm in Table 5.3 reduces to the BiG-AMP algorithm in Table III of [8].

First, we note that the $\widehat{x}_{nl}(t) = \widehat{c}_{\phi_{\mathsf{c}}(n,l)}(t)$ computed by P-BiG-AMP in lines (R2) and (R21) of Table 5.3 is identical to the $\widehat{x}_{nl}(t)$ computed by BiG-AMP in line (R14) of [8, Table III] when $\boldsymbol{X}(\cdot)$ is trivial. Likewise, the $\widehat{a}_{mn}(t) = \widehat{b}_{\phi_{\mathsf{b}}(m,n)}(t)$ computed by P-BiG-AMP in lines (R1) and (R23) of Table 5.3 is identical to the $\widehat{a}_{mn}(t)$ computed by BiG-AMP in line (R16) of [8, Table III] when $\boldsymbol{A}(\cdot)$ is trivial. As for the corresponding variance estimates, $\nu^c_{\phi_{\mathsf{c}}(n,l)}(t)$ computed by P-BiG-AMP in line (R20) of Table 5.3 is identical to the $\nu^x_{nl}(t)$ computed by BiG-AMP in line (R13) of [8, Table III] when $\boldsymbol{X}(\cdot)$ is trivial. Likewise, $\nu^b_{\phi_{\mathsf{b}}(m,n)}(t)$ computed by P-BiG-AMP in line (R22) of Table 5.3 is identical to the $\nu^a_{mn}(t)$ computed by BiG-AMP in line (R15) of [8, Table III] when $\boldsymbol{A}(\cdot)$ is trivial.

[46]The details are implemented in the `A_PLinTrans_X_trivial_ParametricZ` class from the GAMPmatlab package [19].

Next we examine the quantity $\overline{\nu}^p_{ml}(t)$ computed for P-BiG-AMP in line (R9) of Table 5.3. When both $\boldsymbol{X}(\cdot)$ and $\boldsymbol{A}(\cdot)$ are trivial, we can plug (5.161) and a similar identity for $\widehat{z}^{(i,*)}_{ml}(t)$ into (R9) to obtain

$$\overline{\nu}^p_{ml}(t) = \sum_{i=1}^{N_b} \left| \sum_{n=1}^{N} \underbrace{\widehat{a}^{(i)}_{mn}(t)}_{\delta_{i-\phi_{\mathsf{b}}(m,n)}} \widehat{x}_{nl}(t) \right|^2 \nu^b_i(t) + \sum_{j=1}^{N_c} \left| \sum_{n=1}^{N} \widehat{a}_{mn}(t) \underbrace{\widehat{x}^{(j)}_{nl}(t)}_{\delta_{j-\phi_{\mathsf{c}}(n,l)}} \right|^2 \nu^c_j(t) \tag{5.162}$$

$$= \sum_{n=1}^{N} |\widehat{x}_{nl}(t)|^2 \underbrace{\nu^b_{\phi_{\mathsf{b}}(m,n)}(t)}_{=\nu^a_{mn}(t)} + |\widehat{a}_{mn}(t)|^2 \underbrace{\nu^c_{\phi_{\mathsf{c}}(n,l)}(t)}_{=\nu^x_{nl}(t)}. \tag{5.163}$$

For (5.163) we used the facts that, for any fixed combination of $(i,m)$, there is at most one value of $n$ that yields $i = \phi_{\mathsf{b}}(m,n)$, and for any fixed pair $(j,l)$, there is at most one value of $n$ that yields $j = \phi_{\mathsf{c}}(n,l)$. Then, by inspection of (5.163), the $\overline{\nu}^p_{ml}(t)$ computed by P-BiG-AMP is identical to the one computed by BiG-AMP in line (R1) of [8, Table III] when $\boldsymbol{A}(\cdot)$ and $\boldsymbol{X}(\cdot)$ are both trivial.

Regarding the $\nu^p_{ml}(t)$ computed by P-BiG-AMP in line (R10) of Table 5.3, we note that trivial $\boldsymbol{A}(\cdot)$ and $\boldsymbol{X}(\cdot)$ implies

$$\sum_{i=1}^{N_b}\sum_{j=1}^{N_c} \nu^b_i(t)\nu^c_j(t)|\widehat{z}^{(i,j)}_{ml}(t)|^2 = \sum_{i=1}^{N_b}\sum_{j=1}^{N_c} \nu^b_i(t)\nu^c_j(t) \left| \sum_{n=1}^{N} \underbrace{\widehat{a}^{(i)}_{mn}(t)}_{\delta_{i-\phi_{\mathsf{b}}(m,n)}} \underbrace{\widehat{x}^{(j)}_{nl}(t)}_{\delta_{j-\phi_{\mathsf{c}}(n,l)}} \right|^2 \tag{5.164}$$

$$= \sum_{n=1}^{N} \underbrace{\nu^b_{\phi_{\mathsf{b}}(m,n)}(t)}_{=\nu^a_{mn}(t)} \underbrace{\nu^c_{\phi_{\mathsf{c}}(n,l)}(t)}_{=\nu^x_{nl}(t)}, \tag{5.165}$$

since, for any fixed combination of $(i,j,m,l)$, there is at most one value of $n$ that yields $i = \phi_{\mathsf{b}}(m,n)$ and $j = \phi_{\mathsf{c}}(n,l)$. Thus, the $\nu^p_{ml}(t)$ computed by P-BiG-AMP is identical to the one computed by BiG-AMP in line (R3) of [8, Table III] when $\boldsymbol{A}(\cdot)$ and $\boldsymbol{X}(\cdot)$ are both trivial.

By inspection, we can see that the quantity $\widehat{z}^{(*,*)}_{ml}(t)$ computed by P-BiG-AMP in line (R8) of Table 5.3 is identical to $\overline{p}_{ml}(t)$ computed by BiG-AMP in line (R2)

of [8, Table III], and that the quantities $\widehat{p}_{ml}(t), \nu^z_{ml}(t), \widehat{z}_{ml}(t), \nu^s_{ml}(t), \widehat{s}_{ml}(t)$ are identical across algorithms.

Next, with trivial $\boldsymbol{A}(\cdot)$ and $\boldsymbol{X}(\cdot)$, the quantity $\nu^r_j(t)$ computed by P-BiG-AMP in line (R16) of Table 5.3 can be written, using $j = \phi_{\mathsf{c}}(n,l)$ and (5.161), as

$$\nu^r_{\phi_{\mathsf{c}}(n,l)}(t) = \left( \sum_{m,l} \nu^s_{ml}(t) \left| \sum_{k=1}^{N} \widehat{a}_{mk}(t) \underbrace{\widehat{x}^{(\phi_{\mathsf{c}}(n,l))}_{kl}(t)}_{\delta_{\phi_{\mathsf{c}}(n,l)-\phi_{\mathsf{c}}(k,l)}} \right|^2 \right)^{-1} \tag{5.166}$$

$$= \left( \sum_{m,l} \nu^s_{ml}(t) |\widehat{a}_{mn}(t)|^2 \right)^{-1}, \tag{5.167}$$

since, for any pair $(n,l)$, the unique value of $k$ satisfying $\phi_{\mathsf{c}}(n,l) = \phi_{\mathsf{c}}(k,l)$ is $k = n$. From (5.167), it becomes evident that P-BiG-AMP's $\nu^r_{\phi_{\mathsf{c}}(n,l)}(t)$ is identical to BiG-AMP's $\nu^r_{nl}(t)$ from step (R9) of [8, Table III]. Similar arguments show that P-BiG-AMP's $\nu^q_{\phi_{\mathsf{b}}(m,n)}(t)$ is identical to BiG-AMP's $\nu^q_{mn}(t)$ from step (R11) of [8, Table III].

Finally, with trivial $\boldsymbol{A}(\cdot)$ and $\boldsymbol{X}(\cdot)$, the quantity $\widehat{r}_j(t)$ computed by P-BiG-AMP in line (R17) of Table 5.3 can be written, using $j = \phi_{\mathsf{c}}(n,l)$, as

$$\widehat{r}_{\phi_{\mathsf{c}}(n,l)}(t) = \widehat{c}_{\phi_{\mathsf{c}}(n,l)}(t) + \nu^r_{\phi_{\mathsf{c}}(n,l)}(t) \sum_{m=1}^{M} \sum_{k=1}^{L} \left( \widehat{s}_{mk}(t) \widehat{z}^{(*,\phi_{\mathsf{c}}(n,l))}_{mk}(t)^* - \nu^s_{mk}(t) \sum_{i=1}^{N_b} \nu^b_i(t) \widehat{z}^{(i,\phi_{\mathsf{c}}(n,l))}_{mk}(t)^* \widehat{z}^{(i,*)}_{mk}(t) \right) \tag{5.168}$$

Since $\widehat{z}^{(i,\phi_{\mathsf{c}}(n,l))}_{mk}(t) = \delta_{l-k}\delta_{i-\phi_{\mathsf{b}}(m,n)}$, we find that

$$\widehat{r}_{\phi_{\mathsf{c}}(n,l)}(t) = \widehat{c}_{\phi_{\mathsf{c}}(n,l)}(t) + \nu^r_{\phi_{\mathsf{c}}(n,l)}(t) \sum_{m=1}^{M} \left( \widehat{s}_{ml}(t) \widehat{z}^{(*,\phi_{\mathsf{c}}(n,l))}_{ml}(t)^* - \nu^s_{ml}(t) \nu^b_{\phi_{\mathsf{b}}(m,n)}(t) \widehat{z}^{(\phi_{\mathsf{b}}(m,n),*)}_{ml}(t) \right) \tag{5.169}$$

$$= \widehat{x}_{nl}(t) + \underbrace{\nu^r_{\phi_{\mathsf{c}}(n,l)}(t)}_{\nu^r_{nl}(t)} \sum_{m=1}^{M} \left( \widehat{s}_{ml}(t) \widehat{a}^*_{mn}(t) - \nu^s_{ml}(t) \underbrace{\nu^b_{\phi_{\mathsf{b}}(m,n)}(t)}_{\nu^a_{mn}(t)} \widehat{x}_{nl}(t) \right), \tag{5.170}$$

which is identical to $\widehat{r}_{nl}(t)$ computed in line (R10) of [8, Table III]. Similar arguments show that P-BiG-AMP's $\widehat{q}_{\phi_{\mathsf{b}}(m,n)}(t)$ is identical to BiG-AMP's $\widehat{q}_{mn}(t)$ from step (R12) of [8, Table III].

At this point, we have examined all the quantities computed by P-BiG-AMP and thus established that, in the case of trivial $\boldsymbol{A}(\cdot)$ and $\boldsymbol{X}(\cdot)$, the P-BiG-AMP algorithm in Table 5.3 reduces to the BiG-AMP algorithm in Table III of [8].

## 5.7 Implicit Operators

We now discuss the case where some or all of $\widehat{\boldsymbol{A}}(t)$, $\widehat{\boldsymbol{X}}(t)$, $\{\widehat{\boldsymbol{A}}^{(i)}(t)\}_{i=1}^{N_b}$, and $\{\widehat{\boldsymbol{X}}^{(j)}(t)\}_{j=1}^{N_c}$ computed in steps (R1)-(R4) of Table 5.3 can be represented by *implicit* linear operators (from $\mathbb{R}^N \to \mathbb{R}^M$ or $\mathbb{R}^L \to \mathbb{R}^N$) rather than as explicit matrices. As we shall see, the use of implicit operators can lead to significant simplifications in memory and complexity relative to the naive implementation of P-BiG-AMP that executes the algorithm exactly as stated in Table 5.3.

To see how implicit operators might be useful, we first recall from Table 5.3 that $\widehat{\boldsymbol{A}}(t)$, $\widehat{\boldsymbol{X}}(t)$, $\{\widehat{\boldsymbol{A}}^{(i)}(t)\}_{i=1}^{N_b}$, and $\{\widehat{\boldsymbol{X}}^{(j)}(t)\}_{j=1}^{N_c}$ are used only to construct $\widehat{z}_{ml}^{(i,j)}(t)$, $\widehat{z}_{ml}^{(i,*)}(t)$, $\widehat{z}_{ml}^{(*,j)}(t)$, and $\widehat{z}_{ml}^{(*,*)}(t)$, which in turn are used by the remaining steps of the algorithm. Thus, there is no need for explicit computation of $\widehat{\boldsymbol{A}}(t)$, $\widehat{\boldsymbol{X}}(t)$, $\{\widehat{\boldsymbol{A}}^{(i)}(t)\}_{i=1}^{N_b}$, and $\{\widehat{\boldsymbol{X}}^{(j)}(t)\}_{j=1}^{N_c}$. Furthermore, $\widehat{z}_{ml}^{(i,j)}(t)$, $\widehat{z}_{ml}^{(i,*)}(t)$, $\widehat{z}_{ml}^{(*,j)}(t)$ are used only to construct $\overline{\nu}_{ml}^{p}(t), \nu_{ml}^{p}(t), \nu_{j}^{r}(t), \widehat{r}_j(t), \nu_i^q(t), \widehat{q}_i(t)$. Thus, this is no need for explicit computation of $\widehat{z}_{ml}^{(i,j)}(t)$, $\widehat{z}_{ml}^{(i,*)}(t)$, $\widehat{z}_{ml}^{(*,j)}(t)$ either. By avoiding explicit computation of the aforementioned quantities, the complexity and memory burden of P-BiG-AMP can be significantly reduced.

We now propose an alternative implementation strategy for P-BiG-AMP that is able to exploit the advantages of implicit operators, should they exist. In particular, we propose to replace steps (R1)-(R10) with procedure (5.171) below, and to replace

steps (R16)-(R19) with procedure (5.172) below;[47] steps (R11)-(R15) and (R20)-(R23) are to be implemented exactly as described in Table 5.3.

$$\big[\{\widehat{z}_{ml}^{(*,*)}(t)\},\{\overline{\nu}_{ml}^{p}(t)\},\{\nu_{ml}^{p}(t)\}\big] = \mathcal{F}^{p}\big(\widehat{\boldsymbol{b}}(t),\widehat{\boldsymbol{c}}(t),\{\nu_{i}^{b}(t)\},\{\nu_{j}^{c}(t)\}\big) \tag{5.171}$$

$$\big[\{\nu_{j}^{r}(t)\},\widehat{\boldsymbol{r}}(t),\{\nu_{i}^{q}(t)\},\widehat{\boldsymbol{q}}(t)\big] = \mathcal{F}^{rq}\big(\widehat{\boldsymbol{b}}(t),\widehat{\boldsymbol{c}}(t),\{\nu_{i}^{b}(t)\},\{\nu_{j}^{c}(t)\},\widehat{\boldsymbol{S}}(t),\{\nu_{ml}^{s}(t)\}\big). \tag{5.172}$$

The most efficient implementations of $\mathcal{F}^{p}$ and $\mathcal{F}^{rq}$ will be highly dependent on the joint nature of $\boldsymbol{A}(\cdot)$ and $\boldsymbol{X}(\cdot)$. We note, however, that the overall structure of the proposed implementation strategy (5.171)-(5.172) is generic enough to handle any $\boldsymbol{A}(\cdot)$ and $\boldsymbol{X}(\cdot)$. In fact, it can be used even when $\widehat{\boldsymbol{A}}(t)$, $\widehat{\boldsymbol{X}}(t)$, $\{\widehat{\boldsymbol{A}}^{(i)}(t)\}_{i=1}^{N_b}$, and $\{\widehat{\boldsymbol{X}}^{(j)}(t)\}_{j=1}^{N_c}$ are explicit matrices, although in this latter case there would be no advantage over directly calling the steps in Table 5.3.

### 5.7.1 Sandwich $\boldsymbol{A}(\cdot)$ and Trivial $\boldsymbol{X}(\cdot)$

To illustrate the possible advantage of the proposed implementation strategy, we now detail the construction of (5.171)-(5.172) in the case that $\boldsymbol{X}(\cdot)$ is a trivial operator (as discussed in Section 5.6.2) and $\boldsymbol{A}(\cdot)$ is what we call a "sandwich" operator:[48]

$$\widehat{\boldsymbol{A}}(t) = \boldsymbol{A}\big(\widehat{\boldsymbol{b}}(t)\big) = \boldsymbol{F}\underbrace{\boldsymbol{A}_{0}\big(\widehat{\boldsymbol{b}}(t)\big)}_{\triangleq\, \widehat{\boldsymbol{A}}_{0}(t)}\boldsymbol{W}, \tag{5.173}$$

which occurs in a multitude of practical applications. For this, we focus on the interesting case[49] where at least one of the operators $\boldsymbol{F}$, $\boldsymbol{W}$, and $\widehat{\boldsymbol{A}}_{0}(t)$ is implicit,

[47] These two procedures are implemented by objects of the `ParametricZ` class in the MATLAB implementation [19].

[48] In the MATLAB implementation [19], this combination of $\boldsymbol{A}(\cdot)$ and $\boldsymbol{X}(\cdot)$ is handled by the `A_PLinTrans_X_trivial_ParametricZ` subclass of `ParametricZ`.

[49] Notice that this case encompasses arbitrary implicit operators $\widehat{\boldsymbol{A}}(t)$ by setting $\boldsymbol{F}$ and $\boldsymbol{W}$ as appropriately sized identity matrices.

meaning that $\widehat{\boldsymbol{A}}(t)$ and the corresponding derivative operators $\{\widehat{\boldsymbol{A}}^{(i)}(t)\}_{i=1}^{N_b}$ are also implicit.

First, we briefly describe some practical applications that involve this setup. In one example, $\widehat{\boldsymbol{A}}(t)$ is a non-uniform FFT operator (with $\widehat{\boldsymbol{b}}(t)$ specifying the sampling locations), for which multiplication with an $N$-vector can be implemented implicitly with only $O(N \log N)$ complexity [89] and the matrices $\{\widehat{\boldsymbol{A}}^{(i)}(t)\}_{i=1}^{N_b}$ would each be sparse, leading to with $O(N)$-complexity multiply operations. Another example is where $\boldsymbol{W}$ represents a sparsifying basis (such as a wavelet transform), $\boldsymbol{A}_0$ represents a measurement operation with parametric uncertainty, and $\boldsymbol{F}$ represents domain conversion (such as a Fourier transformation) and/or subsampling. A variety of problems in medical imaging, geophysics, and related applications can be cast in this form, and they often feature dimensionalities large enough that explicit construction of $\widehat{\boldsymbol{A}}(t)$ is impractical. Finally, a wide variety of blind deconvolution problems, where $\boldsymbol{A}_0$ represents a convolution in one or more dimensions with an unknown kernel $\widehat{\boldsymbol{b}}(t)$, and where $\boldsymbol{F}$ and $\boldsymbol{W}$ represent domain-specific pre/post-processing operations, can also be modeled using (5.173).

We now turn to the details of implementing $\mathcal{F}^p$ and $\mathcal{F}^{rq}$ for trivial $\boldsymbol{X}(\cdot)$ and sandwich $\boldsymbol{A}(\cdot)$. First, as shown in Section 5.6.2, the quantities $\widehat{z}_{ml}^{(i,j)}(t)$ and $\widehat{z}_{ml}^{(*,j)}(t)$ are sparse versions of elements $\widehat{\boldsymbol{A}}(t)$ and $\{\widehat{\boldsymbol{A}}^{(i)}(t)\}_{i=1}^{N_b}$. It can be shown[50] that the P-BiG-AMP computations involving $\widehat{z}_{ml}^{(i,j)}(t)$ and $\widehat{z}_{ml}^{(*,j)}(t)$ can all be implemented via the implicit operators $\widehat{\boldsymbol{A}}(t)$ and $\{\widehat{\boldsymbol{A}}^{(i)}(t)\}_{i=1}^{N_b}$. Similarly, $\widehat{z}_{ml}^{(*,*)}(t)$ can be computed using implicit $\widehat{\boldsymbol{A}}(t)$. On the other hand, for this particular combination of $\boldsymbol{A}(\cdot)$ and

[50]For these low-level details, as well as other low-level details in this section, we refer the reader to `A_PLinTrans_X_trivial_ParametricZ.m` in [19].

$\boldsymbol{X}(\cdot)$, we find it necessary to explicitly compute and store $\widehat{z}_{ml}^{(i,*)}(t)$ using (5.157) and the implicit operators $\{\widehat{\boldsymbol{A}}^{(i)}(t)\}_{i=1}^{N_b}$.

Considering $\mathcal{F}^p$, we see that steps (R9) and (R10) use the squared entries $|\widehat{z}_{ml}^{(i,j)}(t)|^2$ and $|\widehat{z}_{ml}^{(*,j)}(t)|^2$, which are not explicitly available. In particular, due to the trivial nature of $\boldsymbol{X}(\cdot)$, it can be seen that steps (R9) and (R10) require multiplications with $|\widehat{a}_{mn}(t)|^2$ and $|\widehat{a}_{mn}^{(i)}(t)|^2$, which can be approximated using the Frobenius norm. For example, the second term in (R9) can be written as

$$\sum_{j=1}^{N_c} \widehat{z}_{ml}^{(*,j)}(t)^2 \nu_j^c(t) = \sum_{n=1}^{N} |\widehat{a}_{mn}(t)|^2 \nu_{\phi_{\mathsf{c}}(n,l)}^c(t) \tag{5.174}$$

$$\approx \sum_{n=1}^{N} \left( \frac{1}{MN} \sum_{\overline{m}=1}^{M} \sum_{\overline{n}=1}^{N} |\widehat{a}_{\overline{mn}}(t)|^2 \right) \nu_{\phi_{\mathsf{c}}(n,l)}^c(t) \tag{5.175}$$

$$= \frac{\|\widehat{\boldsymbol{A}}(t)\|_F^2}{MN} \sum_{n=1}^{N} \nu_{\phi_{\mathsf{c}}(n,l)}^c(t), \tag{5.176}$$

where the first step plugs in (5.161) and uses the definition of the trivial mapping for $\boldsymbol{X}(\cdot)$. The required Frobenius norm will be constant and known for some parameterizations, such as the non-uniform FFT. For more general cases, the Frobenius norm can be estimated using a small number of multiplications with random vectors.

Turning to $\mathcal{F}^{rq}$, a similar Frobenius approximation can be made to address the squared entries in step (R16), whereas the squared entries needed for (R18) are available. The remaining issue lies in steps (R17) and (R19). Considering (R17), we see that it requires us to work with terms of the form $\widehat{z}_{ml}^{(i,j)}(t)^* \widehat{z}_{ml}^{(i,*)}(t)$, which we can avoid by making an approximation. Assuming for a moment that $\boldsymbol{A}(\cdot)$ is affine, we

can write

$$\sum_{m,l} \nu^s_{ml}(t) \left( \sum_{i=1}^{N_b} \nu^b_i(t) \widehat{z}^{(i,j)*}_{ml} \widehat{z}^{(i,*)}_{ml}(t) \right)$$

$$= \sum_{m,l} \nu^s_{ml}(t) \left( \sum_{i=1}^{N_b} \nu^b_i(t) \widehat{z}^{(i,j)*}_{ml} \left( \sum_{k=0}^{N_c} \widehat{c}_k \widehat{z}^{(i,k)}_{ml} \right) \right) \tag{5.177}$$

$$= \sum_{i=1}^{N_b} \nu^b_i(t) \sum_{k=0}^{N_c} \widehat{c}_k \left( \sum_{m,l} \nu^s_{ml}(t) \widehat{z}^{(i,j)*}_{ml} \widehat{z}^{(i,k)}_{ml} \right) \tag{5.178}$$

$$\approx \sum_{i=1}^{N_b} \nu^b_i(t) \widehat{c}_j \left( \sum_{m,l} \nu^s_{ml}(t) |\widehat{z}^{(i,j)}_{ml}|^2 \right) \tag{5.179}$$

$$= \widehat{c}_j \sum_{m,l} \nu^s_{ml}(t) \left( \sum_{i=1}^{N_b} \nu^b_i(t) |\widehat{z}^{(i,j)}_{ml}(t)|^2 \right), \tag{5.180}$$

where (5.177) plugs in (5.149), and (5.179) makes the approximation $\sum_{m,l} \nu^s_{ml}(t) \widehat{z}^{(i,j)*}_{ml} \widehat{z}^{(i,k)}_{ml} \approx 0$ for $k \neq j$. We also restore the $t$ dependance on $|\widehat{z}^{(i,j)}_{ml}(t)|^2$ to emphasize that this approximation can be applied for non-affine $\boldsymbol{A}(\cdot)$. The neglected term is a weighted Frobenius inner product between two $\widehat{z}^{(i,j)}_{ml}$ matrices with $\nu^s_{ml}(t)$ acting as the weighting matrix. In the case of trivial $\boldsymbol{X}(\cdot)$ considered here, the term neglected in this approximation reduces to the weighted inner product between two distinct columns of $\widehat{\boldsymbol{A}}^{(i)}(t)$, which is likely small in many applications. With this approximation, step (R17) has been written in terms of $|\widehat{a}^{(i)}_{mn}(t)|^2$ and can now be further approximated using $\|\widehat{\boldsymbol{A}}^{(i)}(t)\|^2_F$, analogously to (5.176), to avoid computing the entry-wise squares. We can obtain a similar approximation for the double-sum term in (R19):

$$\sum_{m,l} \nu^s_{ml}(t) \left( \sum_{j=1}^{N_c} \nu^c_j(t) \widehat{z}^{(i,j)}_{ml}(t)^* \widehat{z}^{(*,j)}_{ml}(t) \right) \approx \widehat{b}_i \sum_{m,l} \nu^s_{ml}(t) \left( \sum_{j=1}^{N_c} \nu^c_j(t) |\widehat{z}^{(i,j)}_{ml}(t)|^2 \right), \tag{5.181}$$

which can also be further approximated using the Frobenius norm.

## 5.8 Adaptive Damping

### 5.8.1 Damping

Damping strategies have been proposed for both GAMP [74] and BiG-AMP [8] to prevent divergence when using matrices that differ from the ideal case of an infinite-dimensional i.i.d sub-Gaussian matrix. For GAMP, damping yields provable local-convergence guarantees with arbitrary matrices [18] while, for BiG-AMP, damping has been shown to be very effective through an extensive empirical study [9].

Motivated by these successes, we adopt a similar damping scheme for P-BiG-AMP. In particular, we use the iteration-$t$ damping factor $\beta(t) \in [0,1]$ to slow the evolution of certain variables, namely, $\overline{\nu}^p_{ml}$, $\nu^p_{ml}$, $\nu^s_{ml}$, $\widehat{s}_{ml}$, $\widehat{b}_i$, and $\widehat{c}_j$. To do this, we replace steps (R9), (R10), (R14), and (R15) in Table 5.3 with

$$\begin{aligned}\overline{\nu}^p_{ml}(t) &= \beta(t)\bigg(\sum_{i=1}^{N_b}|\widehat{z}^{(i,*)}_{ml}(t)|^2\nu^b_i(t) + \sum_{j=1}^{N_c}|\widehat{z}^{(*,j)}_{ml}(t)|^2\nu^c_j(t)\bigg)\\ &\quad + (1-\beta(t))\overline{\nu}^p_{ml}(t-1) \qquad (5.182)\\ \nu^p_{ml}(t) &= \beta(t)\bigg(\overline{\nu}^p_{ml}(t) + \sum_{i=1}^{N_b}\sum_{j=1}^{N_c}\nu^b_i(t)\nu^c_j(t)|\widehat{z}^{(i,j)}_{ml}(t)|^2\bigg)\\ &\quad + (1-\beta(t))\nu^p_{ml}(t-1) \qquad (5.183)\\ \nu^s_{ml}(t) &= \beta(t)\big((1-\nu^z_{ml}(t)/\nu^p_{ml}(t))/\nu^p_{ml}(t)\big)\\ &\quad + (1-\beta(t))\nu^s_{ml}(t-1) \qquad (5.184)\\ \widehat{s}_{ml}(t) &= \beta(t)\big(\widehat{z}_{ml}(t) - \widehat{p}_{ml}(t))/\nu^p_{ml}(t)\big)\\ &\quad + (1-\beta(t))\widehat{s}_{ml}(t-1), \qquad (5.185)\end{aligned}$$

and we insert the following lines between (R15) and (R16):

$$\overline{b}_i(t) = \beta(t)\widehat{b}_i(t) + (1-\beta(t))\overline{b}_i(t-1) \tag{5.186}$$

$$\overline{c}_j(t) = \beta(t)\widehat{c}_j(t) + (1-\beta(t))\overline{c}_j(t-1) \tag{5.187}$$

$$\overline{z}_{ml}^{(i,j)}(t) = \sum_{n=1}^{N} a_{mn}^{(i)}\big(\overline{\boldsymbol{b}}(t)\big)\, x_{nl}^{(j)}\big(\overline{\boldsymbol{c}}(t)\big) \tag{5.188}$$

$$\overline{z}_{ml}^{(i,*)}(t) = \sum_{n=1}^{N} a_{mn}^{(i)}\big(\overline{\boldsymbol{b}}(t)\big)\, x_{nl}\big(\overline{\boldsymbol{c}}(t)\big) \tag{5.189}$$

$$\overline{z}_{ml}^{(*,j)}(t) = \sum_{n=1}^{N} a_{mn}\big(\overline{\boldsymbol{b}}(t)\big)\, x_{nl}^{(j)}\big(\overline{\boldsymbol{c}}(t)\big). \tag{5.190}$$

The quantities $\overline{z}_{ml}^{(i,j)}(t)$, $\overline{z}_{ml}^{(i,*)}(t)$, and $\overline{z}_{ml}^{(*,j)}(t)$ are then used in steps (R16)-(R19), but not in (R9)-(R11), in place of the versions computed in steps (R5)-(R7). The newly created state variables $\overline{b}_i(t)$ and $\overline{c}_j(t)$ are used only to compute these replacements. Notice that, when $\beta(t) = 1$, the damping has no effect, whereas when $\beta(t) = 0$, all quantities become frozen in $t$.

### 5.8.2 Adaptive Damping

Because damping slows the convergence of the algorithm, we would like to damp only as much as needed to prevent divergence. With this in mind, we propose a scheme to adapt $\beta(t)$ by monitoring an appropriate cost $J(t)$. Omitting the derivation details for brevity, the cost we use,

$$\begin{aligned}\widehat{J}(t) = &\sum_j D\Big(p_{\mathsf{c}_j|\mathsf{r}_j}\big(\,\cdot\,\big|\,\widehat{r}_j(t);\nu_j^r(t)\big)\Big\|\,p_{\mathsf{c}_j}(\cdot)\Big) \\ &+ \sum_i D\Big(p_{\mathsf{b}_i|\mathsf{q}_i}\big(\,\cdot\,\big|\,\widehat{q}_i(t);\nu_i^q(t)\big)\Big\|\,p_{\mathsf{b}_i}(\cdot)\Big) \\ &- \sum_{m,l} \mathrm{E}_{\mathsf{z}_{ml}\sim\mathcal{N}(\widehat{z}_{ml}^{(*,*)}(t);\nu_{ml}^p(t))}\big\{\log p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml}\,|\,\mathsf{z}_{ml})\big\},\end{aligned} \tag{5.191}$$

is a natural extension of the MMSE-GAMP cost derived in [20], and reduces to the one used for BiG-AMP stepsize adaptation in [8] when both $\boldsymbol{A}(\cdot)$ and $\boldsymbol{X}(\cdot)$ are

trivial. Intuitively, the first term in (5.191) penalizes the deviation between the (P-BiG-AMP approximated) posterior and the assumed prior on $\mathbf{c}$, the second penalizes the deviation between the (P-BiG-AMP approximated) posterior and the assumed prior on $\mathbf{b}$, and the third term rewards highly likely estimates $\mathbf{Z}$.

For stepsize adaptation, we adopt the approach used for both GAMP and BiG-AMP in the public domain GAMPmatlab implementation [19]. In particular, if the current cost $J(t)$ is not smaller than the largest cost in the most recent `stepWindow` iterations, then the "step" is declared unsuccessful, the damping factor $\beta(t)$ is reduced by the factor `stepDec`, and the step is attempted again. These attempts continue until either the cost criterion decreases or the damping factor reaches `stepMin`, at which point the step is considered successful, or the iteration count exceeds $T_{\max}$ or the damping factor reaches `stepTol`, at which point the algorithm terminates. Otherwise, the step is declared successful, and the damping factor is increased by the factor `stepInc` up to a maximum allowed value `stepMax`.

## 5.9 Tuning of the Prior and Likelihood

### 5.9.1 Expectation Maximization

In order to run P-BiG-AMP, we must specify the prior and likelihood models (5.1) and (5.3). In practice, while a reasonable family of distributions may be dictated by the application, the specific parameters of the distributions will need to be tuned. Building on the approach developed to address this challenge for GAMP [62], which was extended successfully to BiG-AMP in [8], we outline a methodology that takes a given set of P-BiG-AMP priors $\{p_{\mathsf{b}_i}(\cdot;\boldsymbol{\theta}), p_{\mathsf{c}_j}(\cdot;\boldsymbol{\theta}), p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml}|\cdot;\boldsymbol{\theta})\}_{\forall m,n,l}$ and tunes

the vector $\boldsymbol{\theta}$ using an expectation-maximization (EM) [61] based approach, with the goal of maximizing its likelihood, i.e., finding $\widehat{\boldsymbol{\theta}} \triangleq \arg\max_{\boldsymbol{\theta}} p_{\mathbf{Y}}(\boldsymbol{Y};\boldsymbol{\theta})$.

Taking $\mathbf{b}$, $\mathbf{c}$, and $\mathbf{Z}$ to be the hidden variables, the EM recursion can be written as [61]

$$\begin{aligned}\widehat{\boldsymbol{\theta}}^{k+1} &= \arg\max_{\boldsymbol{\theta}} \mathrm{E}\left\{ \log p_{\mathbf{b},\mathbf{c},\mathbf{Z},\mathbf{Y}}(\mathbf{b},\mathbf{c},\mathbf{Z},\mathbf{Y};\boldsymbol{\theta}) \,\middle|\, \boldsymbol{Y};\widehat{\boldsymbol{\theta}}^{k} \right\} \\ &= \arg\max_{\boldsymbol{\theta}} \Bigg\{ \sum_i \mathrm{E}\left\{ \log p_{\mathsf{b}_i}(\mathsf{b}_i;\boldsymbol{\theta}) \,\middle|\, \boldsymbol{Y};\widehat{\boldsymbol{\theta}}^{k} \right\} \\ &\quad + \sum_j \mathrm{E}\left\{ \log p_{\mathsf{c}_j}(\mathsf{c}_j;\boldsymbol{\theta}) \,\middle|\, \boldsymbol{Y};\widehat{\boldsymbol{\theta}}^{k} \right\} \\ &\quad + \sum_{m,l} \mathrm{E}\left\{ \log p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml} \,|\, \mathsf{z}_{ml};\boldsymbol{\theta}) \,\middle|\, \boldsymbol{Y};\widehat{\boldsymbol{\theta}}^{k} \right\} \Bigg\} \end{aligned} \tag{5.192}$$

where for (5.192) we used the fact $p_{\mathbf{b},\mathbf{c},\mathbf{Z},\mathbf{Y}}(\mathbf{b},\mathbf{c},\mathbf{Z},\mathbf{Y};\boldsymbol{\theta}) = p_{\mathbf{b}}(\mathbf{b};\boldsymbol{\theta})p_{\mathbf{c}}(\mathbf{c};\boldsymbol{\theta})p_{\mathbf{Y}|\mathbf{Z}}(\mathbf{Y}|\mathbf{Z};\boldsymbol{\theta})\, 1_{\mathbf{Z}-\boldsymbol{A}(\mathbf{b})\boldsymbol{X}(\mathbf{c})}$ and the separability of $p_{\mathbf{b}}$, $p_{\mathbf{c}}$, and $p_{\mathbf{Y}|\mathbf{Z}}$. As can be seen from (5.192), knowledge of the marginal posteriors $\{p_{\mathsf{b}_i|\mathbf{Y}}, p_{\mathsf{c}_j|\mathbf{Y}}, p_{\mathsf{z}_{ml}|\mathbf{Y}}\}_{\forall m,n,l}$ is sufficient to compute the EM update. Since the exact marginal posteriors are too difficult to compute, we employ the iteration-$t$ approximations produced by P-BiG-AMP, i.e.,

$$p_{\mathsf{b}_i|\mathbf{Y}}(b_i \,|\, \boldsymbol{Y}) \approx p_{\mathsf{b}_i|\mathsf{q}_i}\big(b_i \,|\, \widehat{q}_i(t);\nu_i^q(t)\big) \tag{5.193}$$

$$p_{\mathsf{c}_j|\mathbf{Y}}(c_j \,|\, \boldsymbol{Y}) \approx p_{\mathsf{c}_j|\mathsf{r}_j}\big(c_j \,|\, \widehat{r}_j(t);\nu_j^r(t)\big) \tag{5.194}$$

$$p_{\mathsf{z}_{ml}|\mathbf{Y}}(z_{ml} \,|\, \boldsymbol{Y}) \approx p_{\mathsf{z}_{ml}|\mathsf{p}_{ml}}\big(z_{ml} \,|\, \widehat{p}_{ml}(t);\nu_{ml}^p(t)\big), \tag{5.195}$$

for suitably large $t$, where the distributions above are defined in (D1)-(D3) of Table 5.3. In addition, we adopt the "incremental" update strategy from [76], where the maximization over $\boldsymbol{\theta}$ is performed one element at a time while holding the others fixed. The remaining details are analogous to the GAMP case, for which we refer the interested reader to [62].

### 5.9.2 Initialization of $\boldsymbol{\theta}$

The EM procedure requires a good initialization of $\boldsymbol{\theta}$, since it guarantees only local maximization of the parameter likelihood. Since the best choice of initialization procedure will be highly dependent on the nature of $p_{\mathsf{b}_i}(\cdot;\boldsymbol{\theta})$, $p_{\mathsf{c}_j}(\cdot;\boldsymbol{\theta})$, and $p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml}|\cdot;\boldsymbol{\theta})$, we only make a few comments in this section.

Arguably the most common model for the output channel is the AWGN model, in which case $p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml}|z_{ml};\nu^w) = \mathcal{N}(y_{ml};z_{ml},\nu^w)$. An straightforward generalization of is the "possibly incomplete AWGN" (PIAWGN) model, where only a subset $\Omega \subset \{1...M\}\times\{1...L\}$ of the output indices are observed, i.e.,

$$p_{\mathsf{y}_{ml}|\mathsf{z}_{ml}}(y_{ml}|z_{ml};\nu^w) = \begin{cases}\mathcal{N}(y_{ml};z_{ml},\nu^w) & (m,l)\in\Omega \\ \delta(y_{ml}) & (m,l)\notin\Omega.\end{cases} \tag{5.196}$$

For the PIAWGN model, we suggest to initialize the noise variance as

$$\nu^w = \frac{\|P_\Omega(\boldsymbol{Y})\|_F^2}{(\mathrm{SNR}^0+1)|\Omega|}, \tag{5.197}$$

where $\mathrm{SNR}^0$ is a guess of the SNR and $P_\Omega(\cdot)$ zeros the entries with indices in $\Omega$ while preserving the rest. In the case that the SNR is completely unknown, we suggest setting $\mathrm{SNR}^0 = 100$. We note that this is the same recommendation given for BiG-AMP [8].

In most cases, the parameters in $\boldsymbol{\theta}$ affecting the $p_{\mathbf{b}}$ and $p_{\mathbf{c}}$ distributions can be initialized based on domain knowledge. That said, there are some general principles that should be taken into account when doing so. For example, since $\boldsymbol{Z} = \boldsymbol{A}(\boldsymbol{b})\boldsymbol{X}(\boldsymbol{c})$, there is a fundamental ambiguity between the gain of the operator $\boldsymbol{A}(\boldsymbol{b})$ and the gain of the operator $\boldsymbol{X}(\boldsymbol{c})$. Thus, when initializing and/or tuning $\boldsymbol{\theta}$, it is usually sufficient to fix the typical size of the elements in $\boldsymbol{A}(\mathbf{b})$ and adjust only the typical size of the elements in $\boldsymbol{X}(\mathbf{c})$. Similar observations were made in the context of BiG-AMP [9],

which (as we have seen) is equivalent to P-BiG-AMP in the case of trivial $\boldsymbol{A}(\cdot)$ and $\boldsymbol{X}(\cdot)$.

Whereas in BiG-AMP the parameters $\boldsymbol{\theta}$ affected the gain of the operators $\mathbf{A}$ and $\mathbf{X}$ directly, with P-BiG-AMP the parameterizations $\boldsymbol{A}(\cdot)$ and $\boldsymbol{X}(\cdot)$ also play a role. To see this more concretely, we first notice that, for affine $\boldsymbol{A}(\cdot)$, the average squared energy-gain of the random operator $\boldsymbol{A}(\mathbf{b})$, i.e., $\mathrm{E}_{\mathbf{b}}\{\|\boldsymbol{A}(\mathbf{b})\|_F^2\}$, can be related to the mean and variance of $\mathbf{b}$ as follows:

$$\mathrm{E}_{\mathbf{b}}\{\|\boldsymbol{A}(\mathbf{b})\|_F^2\} = \sum_m \sum_n \mathrm{E}_{\mathbf{b}} \left\{ \left| \sum_{i=0}^{N_b} \mathsf{b}_i a_{mn}^{(i)} \right|^2 \right\} \tag{5.198}$$

$$= \sum_m \sum_n \left[ \sum_{i=0}^{N_b} \mathrm{E}_{\mathbf{b}}\{|\mathsf{b}_i|^2\} |a_{mn}^{(i)}|^2 + \sum_{i=0}^{N_b} \sum_{k \neq i} a_{mn}^{(i)} a_{mn}^{(k)*} \mathrm{E}_{\mathbf{b}}\{\mathsf{b}_i\} \mathrm{E}_{\mathbf{b}}\{\mathsf{b}_k^*\} \right] \tag{5.199}$$

$$\approx \sum_m \sum_n \left[ \sum_{i=0}^{N_b} \mathrm{E}_{\mathbf{b}}\{|\mathsf{b}_i|^2\} |a_{mn}^{(i)}|^2 \right] \tag{5.200}$$

$$= \sum_{i=0}^{N_b} \left( |\mathrm{E}_{\mathbf{b}}\{\mathsf{b}_i\}|^2 + \mathrm{var}_{\mathbf{b}}\{\mathsf{b}_i\} \right) \|\boldsymbol{A}^{(i)}\|_F^2, \tag{5.201}$$

where the approximation in (5.200) holds in the large-system limit for the random affine parameterizations described in Section 5.2.1, or for generic affine parameterizations when $\{\mathsf{b}_i\}_{i>0}$ are zero-mean. A similar relationship holds between $\mathrm{E}_{\mathbf{c}}\{\|\boldsymbol{X}(\mathbf{c})\|_F^2\}$ and the mean and variance of $\mathsf{c}_i$.

We now give an example of how these relationships can be used for the initialization of $\boldsymbol{\theta}$ in the case of PIAWGN observations, affine $\boldsymbol{A}(\cdot)$, trivial $\boldsymbol{X}(\cdot)$, and sparse i.i.d zero-mean $\mathsf{c}_j$. Suppose that the (initial) $p_{\mathbf{b}}$ has been chosen, and the goal is to select the initialization parameters in $\widehat{\boldsymbol{\theta}}^0$ that affect $p_{\mathbf{c}}$. We suggest to first set $\widehat{\boldsymbol{\theta}}^0$ such that the sparsity rate of $\mathsf{c}_j$ equals $\xi_0$, where $\xi_0$ is usually chosen according to the

phase-transition curve, as in [62]. Then, set $\widehat{\boldsymbol{\theta}}^0$ such that the variance of $\mathsf{c}_j$ equals

$$\nu_0^c = \frac{\left(\frac{ML}{|\Omega|}\right)\|P_\Omega(\boldsymbol{Y})\|_F^2 - ML\nu^w}{\mathrm{E}_{\mathbf{b}}\{\|\boldsymbol{A}(\mathbf{b})\|_F^2\}L\xi_0}, \tag{5.202}$$

where the expectation is taken over $p_{\mathbf{b}}(\cdot;\widehat{\boldsymbol{\theta}}^0)$. Note that (5.201) can be used to evaluate the denominator in (5.202).

## 5.10 Numerical Examples

In this section we present two numerical examples to demonstrate the effectiveness of P-BiG-AMP.

### 5.10.1 Random Affine $\boldsymbol{A}(\cdot)$ with Trivial $\boldsymbol{X}(\cdot)$

Our first example addresses the recovery of a sparse signal $\boldsymbol{c} \in \mathbb{R}^N$ from AWGN-corrupted outputs $\boldsymbol{y}$ of a linear transformation $\boldsymbol{A} \in \mathbb{R}^{M\times N}$, where $\boldsymbol{A}$ has an affine parameterization with unknown coefficients $\boldsymbol{b} \in \mathbb{R}^{N_b}$. In particular,

$$\boldsymbol{y} = \Big(\boldsymbol{A}^{(0)} + \sum_{i=1}^{N_b} b_i \boldsymbol{A}^{(i)}\Big)\boldsymbol{c} + \boldsymbol{w}, \tag{5.203}$$

where $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \nu^w \boldsymbol{I})$ is unknown noise, $[\boldsymbol{A}^{(0)}]_{mn}$ are drawn iid $\mathcal{N}(0, N_b)$ and assumed known, and $[\boldsymbol{A}^{(i)}]_{mn}$ are drawn iid $\mathcal{N}(0,1)$ and assumed known. The parameter vector $\boldsymbol{b}$ is drawn iid $\mathcal{N}(0,1)$, and the signal $\boldsymbol{c}$ is drawn Bernoulli-Gaussian with sparsity rate $\xi \in (0,1]$ and $\mathcal{N}(0,\nu^c)$ non-zero entries. Our objective is jointly estimating $\boldsymbol{c}$ and $\boldsymbol{b}$.

This problem can be cast in the P-BiG-AMP framework using $L = 1$, a trivial parameterization for $\boldsymbol{X}(\cdot)$, and the random affine model (5.6) for $\boldsymbol{A}(\cdot)$. One small difference w.r.t (5.6) is that the entries of $\boldsymbol{A}^{(0)}$ have larger variance than those of $\boldsymbol{A}^{(i)}|_{i>0}$. This change was made for two reasons. First, the larger affine offset avoids

scaling ambiguities, simplifying the interpretation of the results. Second, this model is consistent with numerous applications of interest where the parameters control small perturbations of an otherwise known measurement operator. We will run P-BiG-AMP on this problem with oracle knowledge of all the underlying distributions. EM-P-BiG-AMP will also be tested while using the methods described in Section 5.9 to learn the distributional parameters $\boldsymbol{\theta} = [\nu^w, \xi, \nu^c]$.

The results for our techniques will be compared against a state-of-the-art optimization approach and oracle bounds. Notice that, given the true value of $\mathbf{c}$, the MMSE estimate of the parameter $\mathbf{b}$ can be written in closed form, and we denote this estimator as the $\mathbf{c}$-oracle. Similarly, given knowledge of $\mathbf{b}$ and the support set of $\mathbf{c}$, the MMSE estimator of $\mathbf{c}$ can be written in closed form as well, and we denote this estimator as the $\mathbf{b}$,support-Oracle. For further comparison, we consider the Weighted and Structured Sparsity Cognizant Total Least Squares (WSS-TLS) approach from [27], which can be written for the problem of interest as

$$(\widehat{\boldsymbol{b}}, \widehat{\boldsymbol{c}}) = \underset{\boldsymbol{b},\boldsymbol{c}}{\arg\min} \left\| \Big( \boldsymbol{A}^{(0)} + \sum_{i=1}^{N_b} b_i \boldsymbol{A}^{(i)} \Big) \boldsymbol{c} - \boldsymbol{y} \right\|_2^2 + \nu^w \|\boldsymbol{b}\|_2^2 + \lambda \|\boldsymbol{c}\|_1, \tag{5.204}$$

with the tuning parameter $\lambda$. WSS-TLS solves this problem in an alternating fashion, holding each unknown fixed on alternate iterations. For a fixed $\widehat{\boldsymbol{b}}$, the problem is solved efficiently as a linear program, whereas the update holding $\widehat{\boldsymbol{c}}$ fixed can be solved in closed form. Following the guidance provided by example code from the authors of [27], we run the algorithm for several values of $\lambda \in [0, \|\boldsymbol{A}^{(0)\mathsf{T}} \boldsymbol{y}\|_\infty]$ and report the best performance. Notice that WSS-TLS is run with oracle knowledge of the useful distributional parameters from $\boldsymbol{\theta}$, along with oracle-aided tuning of $\lambda$.

To assess the performance of P-BiG-AMP for this problem setup, we conducted a series of random trials. The signal dimension was fixed at $N = 256$, the size of the

parameter $\mathbf{b}$ was set at $N_b = 10$, and the sparsity of the unknown signal was fixed at $\|\mathbf{c}\|_0 = 10$, with non-zero entry variance $\nu^c = 1$. The AWGN variance $\nu^w$ was selected to enforce SNR $= 20 \log_{10} \|\boldsymbol{y} - \boldsymbol{w}\|_2 / \|\boldsymbol{w}\|_2 = 40$ dB. Finally, the number of measurements $M$ was varied to cover under sampling ratios $M/N$ ranging from 0.1 to 0.9. The results showing the mean NMSE over 10 trials for estimating both $\mathbf{b}$ and $\mathbf{c}$ from the measurements $\boldsymbol{y}$ are shown in Fig. 5.3. P-BiG-AMP obtains accurate solutions with fewer measurements than WSS-TLS, succeeding at $M/N = 0.2$, whereas WSS-TLS only obtains reasonable reconstructions for $M/N \geq 0.3$. Furthermore, P-BiG-AMP nearly matches the oracle bounds for $M/N \geq 0.3$, whereas WSS-TLS, despite oracle-aided tuning of the parameter $\lambda$, continues to exhibit a performance gap of several dB, even at $M/N = 0.9$. EM-P-BiG-AMP very nearly matches the performance of P-BiG-AMP, with the exception of a small gap at $M/N = 0.2$, in spite of the need to learn the underlying distributional parameters $\boldsymbol{\theta}$.

### 5.10.2 Noisy Partial 2D Fourier Measurements of a Sparse Image with Row-wise Phase Errors

Our second example is motivated by the problem of simultaneous sparse imaging and phase error correction in synthetic aperture radar as studied in [90]. Here, we consider a simplified setup that captures the salient features of this application. In particular, we adopt the model

$$\boldsymbol{y} = \boldsymbol{\Phi} \operatorname{diag}(\boldsymbol{b} \otimes \mathbf{1}_N) \boldsymbol{F} \boldsymbol{c} + \boldsymbol{w}, \tag{5.205}$$

where $\boldsymbol{c} \in \mathbb{C}^{N^2}$ contains the vectorized complex-valued pixels in an $N \times N$ 2-D image that we seek to reconstruct, and $\boldsymbol{F} \in \mathbb{C}^{N^2 \times N^2}$ implements a 2D Discrete Fourier Transform (DFT) with vectorized input and output. The entries of $\boldsymbol{b} \in \mathbb{C}^N$ are

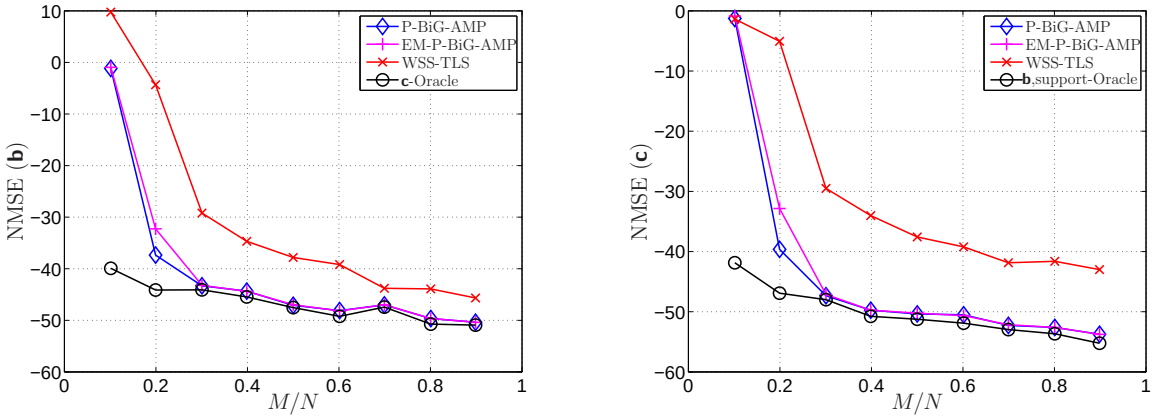


Figure 5.3: NMSE for estimation of the trivially-parameterized sparse signal $\boldsymbol{X}(\mathbf{c}) \in \mathbb{R}^N$ (right) with $\|\mathbf{c}\|_0 = 10$ and the parameter $\mathbf{b} \in \mathbb{R}^{10}$ (left) of the random affine measurement operator $\boldsymbol{A}(\mathbf{b}) \in \mathbb{R}^{M \times N}$ as a function of the ratio between the number of measurements $M$ and the signal length $N = 256$. The measurements were corrupted with AWGN at a SNR of 40 dB. All results represent mean performance over 10 random trials.

drawn uniformly at random on the unit circle in the complex plane. Multiplication with $\text{diag}(\boldsymbol{b} \otimes \mathbf{1}_N)$ thus adds an unknown phase error to all of the Fourier measurements such that the error is constant for each row of the 2D DFT. Finally, $\Phi \in \mathbb{R}^{M \times N^2}$ is a selection matrix which down-samples the phase-corrupted Fourier measurements, and $\boldsymbol{w} \in \mathbb{C}^M$ represents additive circular complex Gaussian noise. This problem also fits into the P-BiG-AMP framework with $\boldsymbol{X}(\cdot)$ chosen as the trivial parameterization, $\boldsymbol{A}(\boldsymbol{b}) = \boldsymbol{\Phi}\,\text{diag}(\boldsymbol{b} \otimes \mathbf{1}_N)\boldsymbol{F}$, and a CAWGN likelihood. The prior on $\mathbf{b}$ can be implemented in the P-BiG-AMP framework using a similar methodology to the one applied for uncertain output phases in [74], while a Bernoulli-Gaussian prior is selected for $\mathbf{c}$. Notice that the model here includes additional structure beyond the standard phase-retrieval problem that P-BiG-AMP is able to exploit, since the unknown phase errors are constant across rows of the DFT output.

We simulated an example with $N = 128$, 300 non-zero pixels with values drawn iid $\mathcal{CN}(0,1)$, $M = 0.2N^2$, phase errors $\boldsymbol{b}_i$ chosen uniformly on $[-90°, 90°]$, and $\nu^w$ selected to obtain SNR = 40 dB. The results are depicted in Fig. 5.4. The top left pane shows the magnitudes of the pixels in the original image on a normalized dB scale. Naively applying GAMP with the correct signal model, with the exception of ignoring the phase errors, produces the severely corrupted result shown on the top right. On the other hand, applying P-BiG-AMP allows the phase errors to be accurately reconstructed, as shown on the bottom right, and simultenously produces an accurate estimate of the original image, shown on the bottom left. The NMSE in the P-BiG-AMP estimate of the image is $-49.62$ dB, rendering the result visually indistinguishable from the original in this example.

## 5.11 Conclusion

In this chapter, we presented P-BiG-AMP, a parametric extension of the BiG-AMP algorithm described in Chapter 4. P-BiG-AMP offers a framework for solving a variety of bilinear inference problems where one or both of the matrix factors has a parsimonious, parametric representation. This parametric approach allowed us to relax the assumption of separable priors on these factors and offers the potential to apply bilinear AMP based inference to a variety of practical problems in remote sensing and other domains. Building on techniques developed for GAMP and BiG-AMP, we also proposed adaptive damping and parameter tuning schemes for the algorithm. Several simplifications and approximations of P-BiG-AMP were considered for particular choices of the parameterizations. Finally, we presented two numerical examples to demonstrate the effectiveness of P-BiG-AMP.

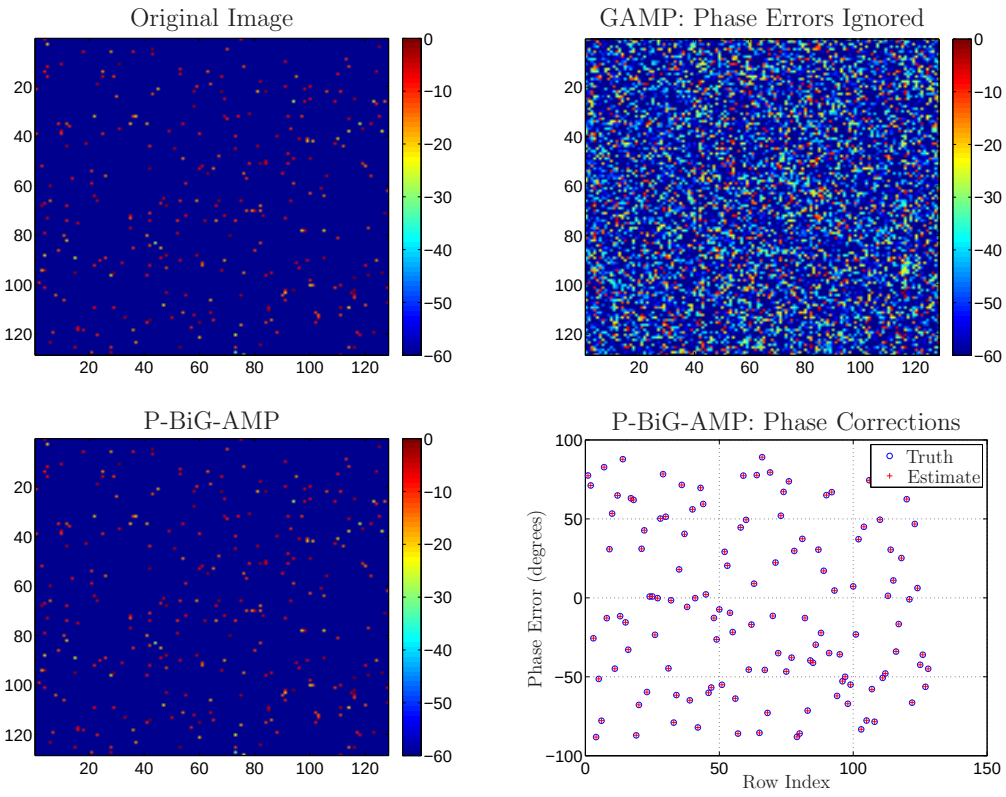


Figure 5.4: Recovery of a $128 \times 128$ complex-valued image with 300 non-zero pixels from partial noisy 2D Fourier measurements with row-wise phase errors uniformly distributed on $[-90°, 90°]$ and AWGN at SNR $= 40$ dB. The magnitude of the original image on a normalized dB scale is shown on the top left, and the reconstruction using GAMP, which ignores the phase errors, is shown on the top right. The P-BiG-AMP estimates of the image and the required phase corrections are shown on the bottom left and right, respectively. The NMSE in the recovery of the image using P-BiG-AMP is $-49.62$ dB.

# Chapter 6: Conclusion and Future Work

Motivated by the success of Compressive Sensing (CS) techniques for solving linear inference problems with sparsity constraints, and in particular by the effectiveness of Approximate Message Passing (AMP) methods, this dissertation has explored several extensions to the Generalized AMP (GAMP) algorithm, enabling its use to solve a variety of bilinear inference problems. First, we developed a Matrix Uncertain GAMP (MU-GAMP) to address measurement matrix uncertainty in problems reminiscent of those considered in the traditional CS literature. Along the way, we demonstrated that uniform uncertainty in the measurement matrix could be handled without specialized algorithms for sufficiently large problems and introduced an alternating version of MU-GAMP that could estimate the measurement matrix itself under certain conditions, empirically demonstrating near-oracle performance. Building on the success of this alternating scheme, a Bilinear GAMP (BiG-AMP) algorithm was derived to jointly estimate both factors of an unknown matrix product. The two matrix factors were modeled using separable priors on their entries, and the resulting algorithm achieved performance comparable, and in some cases superior, to state-of-the-art algorithms on problems in matrix completion, robust principal components analysis, and dictionary learning. A variety of modifications to the algorithm were also proposed and tested to reduce computational cost, learn unknown prior parameters and

model orders, and improve performance on real-world data sets. Finally, an extension of BiG-AMP enabled the use of non-separable priors on the matrix factors by representing them with known parametric models. While the resulting P-BiG-AMP subsumes BiG-AMP for a particular choice of the parameterizations and is justified rigorously for random affine parameterizations, it can be applied to a much wider class of nonlinear parameterizations, offering a host of potential applications.

A variety of directions could be pursued in future work. First, a detailed analysis of the convergence behavior of EM-BiG-AMP under both matched and mismatched statistics would offer insights into the applicability of the algorithm and might lead to improved schemes for adaptive damping. Various model extensions, including structured-sparsity or the ability to impose additional, possibly non-linear constraints on the resulting solutions might allow wider application of BiG-AMP. For example, certain motion estimation problems require the further restriction that the columns of low rank solutions lie on the surface of a low-dimensional ellipsoid. As briefly mentioned in Chapter 4, a variety of practical problems in high-dimensional inference could also be attacked using BiG-AMP, including extensions to preliminary results for hyperspectral unmixing and other problem domains in remote sensing. The most promising future work, however, lies in applying P-BiG-AMP to these and similar problems. Several potential applications are possible, including simultaneous calibration and sparse imaging from Fourier data, registration of sensor data across modalities, automatic learning of sensitivity maps in parallel magnetic resonance imaging, various formulations of blind deconvolution in communications, and numerous others.

# Bibliography


[1] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.

[2] E. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[3] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing," *Proc. Nat. Acad. Sci.*, vol. 106, no. 45, pp. 18 914–18 919, Nov. 2009.

[4] ——, "Message passing algorithms for compressed sensing: I. Motivation and construction," in *Proc. Inform. Theory Workshop*, Cairo, Egypt, Jan. 2010, pp. 1–5.

[5] ——, "Message passing algorithms for compressed sensing: II. Analysis and validation," in *Proc. Inform. Theory Workshop*, Cairo, Egypt, Jan. 2010, pp. 1–5.

[6] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," in *Proc. IEEE Int. Symp. Inform. Thy.*, Saint Petersburg, Russia, Aug. 2011, pp. 2168–2172, (Full version at *arXiv:1010.5141*).

[7] J. T. Parker, V. Cevher, and P. Schniter, "Compressive sensing under matrix uncertainties: An approximate message passing approach," in *Proc. Asilomar Conf. Signals Syst. Comput.*, Pacific Grove, CA, Nov. 2011, pp. 804–808.

[8] J. T. Parker, P. Schniter, and V. Cevher, "Bilinear generalized approximate message passing, Part 1: Derivation," *Submitted to IEEE Trans. Signal Process.*, 2013.

[9] ——, "Bilinear generalized approximate message passing, Part 2: Applications," *Submitted to IEEE Tran. Signal Process.*, 2013.

[10] J. T. Parker and P. Schniter, "Parametric bilinear generalized approximate message passing," *In Preparation for IEEE Trans. Signal Process.*, 2014.

[11] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," *arXiv:1001.3448*, Jan. 2010.

[12] D. Guo and C.-C. Wang, "Asymptotic mean-square optimality of belief propagation for sparse linear systems," in *Proc. Inform. Theory Workshop*, Chengdu, China, Oct. 2006, pp. 194–198.

[13] ——, "Random sparse linear systems observed via arbitrary channels: A decoupling principle," in *Proc. IEEE Int. Symp. Inform. Thy.*, Nice, France, Jun. 2007, pp. 946–950.

[14] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. San Mateo, CA: Morgan Kaufman, 1988.

[15] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[16] A. Montanari, "Graphical models concepts in compressed sensing," in *Compressed Sensing: Theory and Applications*, Y. C. Eldar and G. Kutyniok, Eds. Cambridge Univ. Press, 2012.

[17] 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. 2012, no. 08, p. P08009, 2012.

[18] S. Rangan, P. Schniter, and A. Fletcher, "On the convergence of generalized approximate message passing with arbitrary matrices," in *Proc. IEEE Int. Symp. Inform. Thy.*, Honolulu, Hawaii, Jul. 2014, to appear., (Full version at *arXiv:1402.3210*).

[19] S. Rangan, J. T. Parker, P. Schniter, J. Ziniel, J. Vila, M. Borgerding, and et al., "GAMPmatlab," `https://sourceforge.net/projects/gampmatlab/`.

[20] S. Rangan, P. Schniter, E. Riegler, A. Fletcher, and V. Cevher, "Fixed points of generalized approximate message passing with arbitrary matrices," in *Proc. IEEE Int. Symp. Inform. Thy.*, Istanbul, Turkey, Jul. 2013, pp. 664–668, (Full version at *arXiv:1301.6295*).

[21] V. Cevher, "On accelerated hard thresholding methods for sparse approximation," Idiap Research Institute, Ecole Polytechnique, Tech. Rep., 2011.

[22] L. Anitori, A. Maleki, M. Otten, R. Baraniuk, and P. Hoogeboom, "Design and analysis of compressed sensing radar detectors," *Signal Processing, IEEE Transactions on*, vol. 61, no. 4, pp. 813–827, 2013.

[23] E. Candès, "The restricted isometry property and its implications for compressed sensing," *Comptes rendus-Mathématique*, vol. 346, no. 9-10, pp. 589–592, 2008.

[24] M. A. Herman and T. Strohmer, "General deviants: An analysis of perturbations in compressed sensing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 342–349, 2010.

[25] M. Herman and D. Needell, "Mixed operators in compressed sensing," in *Information Sciences and Systems (CISS), 2010 44th Annual Conference on*. IEEE, 2010, pp. 1–6.

[26] Y. Chi, L. L. Scharf, A. Pezeshki, and A. R. Calderbank, "Sensitivity to basis mismatch in compressed sensing," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2182–2195, May 2011.

[27] H. Zhu, G. Leus, and G. Giannakis, "Sparsity-cognizant total least-squares for perturbed compressive sampling," *Signal Processing, IEEE Transactions on*, vol. 59, no. 5, pp. 2002 –2016, May 2011.

[28] M. Rosenbaum and A. Tsybakov, "Sparse recovery under matrix uncertainty," *The Annals of Statistics*, vol. 38, no. 5, pp. 2620–2651, 2010.

[29] E. Candes and T. Tao, "The dantzig selector: Statistical estimation when p is much larger than n," *The Annals of Statistics*, vol. 35, no. 6, pp. 2313–2351, 2007.

[30] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[31] Y. Tang, L. Chen, and Y. Gu, "On the performance bound of sparse estimation with sensing matrix perturbation," *Signal Processing, IEEE Transactions on*, vol. 61, no. 17, pp. 4372–4386, Sept 2013.

[32] S. Wright, R. Nowak, and M. Figueiredo, "Sparse reconstruction by separable approximation," *Signal Processing, IEEE Transactions on*, vol. 57, no. 7, pp. 2479 –2493, july 2009.

[33] I. Tošić and P. Frossard, "Dictionary learning," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 27–38, Mar. 2011.

[34] J. Cai, E. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/080738970

[35] S. Ma, D. Goldfarb, and L. Chen, "Fixed point and bregman iterative methods for matrix rank minimization," *Mathematical Programming*, vol. 128, no. 1-2, pp. 321–353, 2011. [Online]. Available: http://dx.doi.org/10.1007/s10107-009-0306-5

[36] Z. Zhou, X. Li, J. Wright, E. Candès and, and Y. Ma, "Stable principal component pursuit," in *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*, june 2010, pp. 1518 –1522.

[37] B. Recht, M. Fazel, and P. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Review*, vol. 52, no. 3, pp. 471–501, 2010. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/070697835

[38] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *Arxiv preprint arXiv:1009.5055*, 2010.

[39] M. Tao and X. Yuan, "Recovering low-rank and sparse components of matrices from incomplete and noisy observations," *SIAM Journal on Optimization*, vol. 21, no. 1, pp. 57–81, 2011.

[40] T. Zhou and D. Tao, "Godec: Randomized low-rank and sparse matrix decomposition in noisy case," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ser. ICML '11, L. Getoor and T. Scheffer, Eds. New York, NY, USA: ACM, June 2011, pp. 33–40.

[41] H. Ghasemi, M. Malek-Mohammadi, M. Babaei-Zadeh, and C. Jutten, "SRF: Matrix completion based on smoothed rank function," in *Proc. IEEE Int. Conf. Acoust. Speech & Signal Process.*, Prague, Czech Republic, May 2011.

[42] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Mathematical Programming Computation*, vol. 4, pp. 333–361, 2012. [Online]. Available: http://dx.doi.org/10.1007/s12532-012-0044-1

[43] A. Kyrillidis and V. Cevher, "Matrix recipes for hard thresholding methods," *arXiv:1203.4481*, 2012.

[44] G. Marjanovic and V. Solo, "On optimization and matrix completion," *Signal Processing, IEEE Transactions on*, vol. 60, no. 11, pp. 5714–5724, 2012.

[45] J. Haldar and D. Hernando, "Rank-constrained solutions to linear matrix equations using PowerFactorization," *Signal Processing Letters, IEEE*, vol. 16, no. 7, pp. 584 –587, july 2009.

[46] L. Balzano, R. Nowak, and B. Recht, "Online identification and tracking of subspaces from highly incomplete information," *Arxiv preprint arXiv:1006.4046*, 2010.

[47] R. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," *Information Theory, IEEE Transactions on*, vol. 56, no. 6, pp. 2980 –2998, june 2010.

[48] W. Dai and O. Milenkovic, "SET: An algorithm for consistent matrix completion," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, 2010, pp. 3646–3649.

[49] J. He, L. Balzano, and J. Lui, "Online robust subspace tracking from partial information," *Arxiv preprint arXiv:1109.3827*, 2011.

[50] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," *Advances in neural information processing systems*, vol. 20, pp. 1257–1264, 2008.

[51] X. Ding, L. He, and L. Carin, "Bayesian robust principal component analysis," *Image Processing, IEEE Transactions on*, vol. 20, no. 12, pp. 3419 –3430, dec. 2011.

[52] Y. Lim and Y. Teh, "Variational bayesian approach to movie rating prediction," in *Proceedings of KDD Cup and Workshop*. Citeseer, 2007, pp. 15–21.

[53] S. D. Babacan, M. Luessi, R. Molina, and A. K. Katsaggelos, "Sparse Bayesian methods for low-rank matrix estimation," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 3964–3977, Aug. 2012.

[54] M. Tipping and C. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[55] F. Léger, G. Yu, and G. Sapiro, "Efficient matrix completion with gaussian models," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1113–1116.

[56] N. Wang, T. Yao, J. Wang, and D. Yeung, "A probabilistic approach to robust matrix factorization," in *Proceedings of European Conference on Computer Vision*, A. Fitzgibbon, Ed., vol. VII, no. 2012, 2012, pp. 126–139.

[57] B.-H. Kim, A. Yedla, and H. Pfister, "Imp: A message-passing algorithm for matrix completion," in *Turbo Codes and Iterative Information Processing (ISTC), 2010 6th International Symposium on*, 2010, pp. 462–466.

[58] B. J. Frey and D. J. C. MacKay, "A revolution: Belief propagation in graphs with cycles," in *Proc. Neural Inform. Process. Syst. Conf.*, Denver, CO, 1997, pp. 479–485.

[59] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," *IEEE Trans. Inform. Theory*, vol. 57, no. 2, pp. 764–785, Feb. 2011.

[60] A. Javanmard and A. Montanari, "State evolution for general approximate message passing algorithms, with applications to spatial coupling," *Information and Inference*, Oct. 2013.

[61] A. Dempster, N. M. Laird, and D. B. Rubin, "Maximum-likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc.*, vol. 39, pp. 1–17, 1977.

[62] J. P. Vila and P. Schniter, "Expectation-maximization Gaussian-mixture approximate message passing," *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4658–4672, Oct. 2013.

[63] P. Schniter, "Turbo reconstruction of structured sparse signals," in *Proc. Conf. Inform. Science & Syst.*, Princeton, NJ, Mar. 2010, pp. 1–6.

[64] J. Vila, P. Schniter, and J. Meola, "Hyperspectral image unmixing via bilinear generalized approximate message passing," *Proc. SPIE*, vol. 8743, p. Y, 2013.

[65] P. Schniter and V. Cevher, "Approximate message passing for bilinear models," in *Proc. Workshop Signal Process. Adaptive Sparse Struct. Repr. (SPARS)*, Edinburgh, Scotland, Jun. 2011, p. 68.

[66] P. Schniter, J. T. Parker, and V. Cevher, "Bilinear generalized approximate message passing (BiG-AMP) for matrix recovery problems," Feb. 2012, presented at the *Infom. Thy. & Appl. Workshop (ITA)*, (La Jolla, CA).

[67] S. Rangan and A. K. Fletcher, "Iterative estimation of constrained rank-one matrices in noise," *arXiv:1202.2759*, February 2012.

[68] F. Krzakala, M. Mézard, and L. Zdeborová, "Phase diagram and approximate message passing for blind calibration and dictionary learning," *arXiv:1301.5898*, January 2013.

[69] G. F. Cooper, "The computational complexity of probabilistic inference using Bayesian belief networks," *Artificial Intelligence*, vol. 42, pp. 393–405, 1990.

[70] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning low-level vision," *Intl. J. Computer Vision*, vol. 40, no. 1, pp. 25–47, Oct. 2000.

[71] R. J. McEliece, D. J. C. MacKay, and J.-F. Cheng, "Turbo decoding as an instance of Pearl's 'belief propagation' algorithm," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 140–152, Feb. 1998.

[72] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. New York: Cambridge University Press, 2003.

[73] J. Boutros and G. Caire, "Iterative multiuser joint decoding: Unified framework and asymptotic analysis," *IEEE Trans. Inform. Theory*, vol. 48, no. 7, pp. 1772–1793, Jul. 2002.

[74] P. Schniter and S. Rangan, "Compressive phase retrieval via generalized approximate message passing," in *Proc. Allerton Conf. Commun. Control Comput.*, Monticello, IL, Oct. 2012.

[75] S. Wright, R. Nowak, and M. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2479–2493, Jul. 2009.

[76] R. Neal and G. 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.

[77] P. Stoica and Y. Selen, "Model-order selection: a review of information criterion rules," *Signal Processing Magazine, IEEE*, vol. 21, no. 4, pp. 36 – 47, july 2004.

[78] E. J. Candès and Y. Plan, "Matrix completion with noise," *Proc. IEEE*, vol. 98, no. 6, pp. 925–936, Jun. 2010.

[79] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *J. Roy. Statist. Soc. B*, vol. 61, no. 3, pp. 611–622, 1999.

[80] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. Neural Inform. Process. Syst. Conf.*, Vancouver, BC, Dec. 2008, pp. 1257–1264.

[81] N. D. Lawrence and R. Urtasun, "Non-linear matrix factorization with Gaussian processes," in *Proc. Int. Conf. Mach. Learning*, Montreal, Quebec, Jun. 2009, pp. 601–608.

[82] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 11:1–11:37, Jun. 2011. [Online]. Available: http://doi.acm.org/10.1145/1970392.1970395

[83] R. Rubinstein, A. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1045–1057, 2010.

[84] D. A. Spielman, H. Wang, and J. Wright, "Exact recovery of sparsely-used dictionaries," in *JMLR: Workshop and Conference Proceedings of 25th Annual Conference on Learning Theory*, vol. 23, 2012.

[85] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *Signal Processing, IEEE Transactions on*, vol. 54, no. 11, pp. 4311–4322, 2006.

[86] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *The Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.

[87] P. de Jong, "A central limit theorem for generalized quadratic forms," *Probab. Th. Rel. Fields*, vol. 75, pp. 261–277, 1987.

[88] H. W. Sorenson, *Kalman Filtering: Theory and Application.* Piscataway, NJ: IEEE, 1985.

[89] L. Greenbard and J.-Y. Lee, "Accelerating the nonuniform fast fourier transform," *SIAM Review*, vol. 46, no. 3, pp. 443–454, 2004.

[90] N. Onhon and M. Cetin, "A sparsity-driven approach for joint sar imaging and phase error correction," *Image Processing, IEEE Transactions on*, vol. 21, no. 4, pp. 2075 –2088, april 2012.