Bilinear Generalized Approximate Message Passing

Jason T. Parker, Philip Schniter, and Volkan Cevher

Abstract—We extend the generalized approximate message passing (G-AMP) approach, originally proposed for high-dimensional generalized-linear regression in the context of compressive sensing, to the bilinear-generalized case, which enables its application to matrix completion, robust PCA, dictionary learning, and related matrix-factorization problems. In the first part of the paper, we derive our Bilinear G-AMP (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. In the second part of the paper, we 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.

I. INTRODUCTION

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

\[
p_A(A) = \prod_m \prod_n p_{a_{mn}}(a_{mn}) \tag{1}
\]

\[
p_X(X) = \prod_n \prod_l p_{x_{nl}}(x_{nl}) \tag{2}
\]

and we likewise assume that the likelihood function of \( Z \) is known and separable, i.e.,

\[
p_Y(Z \mid Z_m) = \prod_m \prod_l p_{y_{ml}}(y_{ml} \mid z_{ml}) \tag{3}
\]

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, \( Z = AX \) would represent the complete low-rank matrix (with tall \( A \) and wide \( X \)) and \( p_{y_{ml}}(y_{ml} \mid z_{ml}) \) the observation mechanism, which would be (partially) informative about \( z_{ml} \) 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, \( Z = AX \) could again represent the low-rank matrix, and \( p_{y_{ml}}(y_{ml} \mid z_{ml}) \) the noise-and-outlier-corrupted observation mechanism. Alternatively, \( X \) could also capture the outliers, as described in the sequel.

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

While a plethora of approaches to these problems have been proposed based on optimization techniques (e.g., [4]–[14]), greedy methods (e.g., [15]–[19]), Bayesian sampling methods (e.g., [20], [21]), variational methods (e.g., [22]–[26]), and discrete message passing (e.g., [27]), ours is based on the Approximate Message Passing (AMP) framework, an instance of loopy belief propagation (LBP) [28] that was recently developed to tackle linear [29]–[31] and generalized linear [32] inference problems encountered in the context of compressive sensing (CS). In the generalized-linear CS problem, one estimates \( x \in \mathbb{R}^N \) from observations \( y \in \mathbb{R}^M \) that are statistically coupled to the transform outputs \( z = Ax \) through a separable likelihood function \( p_Y(y | z) \), where in this case the transform \( A \) 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 \( A \)) are governed by a state-evolution whose fixed points—when unique—yield the true posterior means [33, 34] and ii) a low implementation complexity (i.e., dominated by one multiplication with \( A \) and \( A^T \) per iteration, and relatively few iterations) [31]. Thus, a natural question is
whether the AMP framework can be successfully applied to the generalized bilinear problem described earlier.

In this manuscript, we propose an AMP-based approach to generalized bilinear inference that we henceforth refer to as BiLinear Generalized AMP (BiG-AMP), and we uncover cases under which the general approach can be simplified. In addition, we propose an adaptive damping [35] mechanism, an expectation-maximization (EM)-based [36] method of tuning the parameters of $p_{mn}$, $p_{ni}$, and $p_{m|z_{ni}}$ (in case they are unknown), and methods to select the rank $N$ (in case it is unknown). In the case that $p_{mn}$, $p_{ni}$, and/or $p_{m|z_{ni}}$ are completely unknown, they can be modeled as Gaussian-mixtures with mean/variance/weight parameters learned via EM [37]. 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 compared to existing state-of-the-art algorithms for each application.

Although the AMP methodology is itself restricted to separable random variables [1-3], 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” [38] and applied to BiG-AMP in [39]. 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 [31, 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 [1, 2], Rangan and Fletcher [40] 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 [41] 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 [41]. The BiG-AMP algorithm that we derive here is a generalization of those in [40], [41] 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 [39].

The remainder of the document is organized as follows. Section II derives the BiG-AMP algorithm, and Sec. III presents several special-case simplifications of BiG-AMP. Section IV describes the adaptive damping mechanism, and Sec. V 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 VI, VII, VIII respectively, and concluding remarks are offered in Sec. IX.

Notation: Throughout, we use san-serif font (e.g., $X$) for random variables and serif font (e.g., $x$) otherwise. We use boldface capital letters (e.g., $X$ and $X$) for matrices, boldface small letters (e.g., $x$ and $x$) for vectors, and non-bold small letters (e.g., $x$ and $x$) for scalars. We then use $p(x)$ to denote the pdf of random quantity $x$, and $\mathcal{N}(x; \bar{x}, \nu^2)$ to denote the Gaussian pdf for a scalar random variable with mean $\bar{x}$ and variance $\nu^2$. Also, we use $E\{x\}$ and $\text{var}\{x\}$ to denote mean and variance of $x$, respectively, and $\mathcal{D}(p_1\|p_2)$ for the Kullback-Leibler (KL) divergence between pdfs $p_1$ and $p_2$. For a matrix $X$, we use $x_{ni} = [X]_{ni}$ to denote the entry in the $n^{th}$ row and $i^{th}$ column, $\|X\|_F$ to denote the Frobenius norm, and $X^T$ to denote transpose. Similarly, we use $x_n$ to denote the $n^{th}$ entry in vector $x$ and $\|x\|_p = (\sum |x_n|^p)^{1/p}$ to denote the $\ell_p$ norm.

II. BILINEAR GENERALIZED AMP

A. Problem Formulation

For the statistical model [1-3], the posterior distribution is

$$p_{X,A|Y}(X,A|Y) = p_{Y,X,A}(Y|X,A)p_{X}(X)p_{A}(A)/p_{Y}(Y)$$

$$\propto p_{Y,Z}(Y|AX)p_{X}(X)p_{A}(A)$$

$$= \left[\prod_{m} \prod_{l} p_{y_{ml}|z_{ml}} y_{ml} \sum_{a_{ml}} a_{ml} x_{nl} \right] \times \left[\prod_{n} \prod_{l} p_{x_{nl}|z_{ml}} x_{nl} \left[\prod_{m} \prod_{l} p_{a_{ml}|z_{ml}} a_{ml} \right]\right].$$

where [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. I. There, the factors of $p_{X,A|Y}$ from (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 $p_{y_{ml}|z_{ml}}$ factor nodes in the middle of the graph, and not as random variables. The structure of Fig. I becomes intuitive when recalling that $Z = AX$ implies $z_{ml} = \sum_{n=1}^{N} a_{ml} x_{nl}$.

B. Loopy Belief Propagation

In this work, we aim to compute minimum mean-squared error (MMSE) estimates of $X$ and $A$, i.e., the mean of

1Another worthwhile objective could be to compute the joint MAP estimate $\arg \max X,A p_{X,A|Y}(X,A|Y)$; we leave this to future work.
the marginal posteriors \( p_{nk\,l}(\cdot \mid Y)\) and \( p_{mn\,l}(\cdot \mid Y)\), 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) \([28]\).

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) \([42],\ [43]\), 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 nodes, 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 \([42],\ [43]\). For loopy factor graphs, exact inference is in general NP hard \([44]\) 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 \([45]\), turbo decoding \([46]\), LDPC decoding \([47]\), multiuser detection \([48]\), and compressive sensing \([29],\ [30],\ [32]–\[34]\).

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 \([32]\) to solve the generalized CS problem, which exploits the “blessings of dimensionality” that arise when \( A \) is a sufficiently large and dense and which was rigorously analyzed in \([34]\). 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. 1.

As we shall see, the approximations are primarily based on central-limit-theorem (CLT) and Taylor-series arguments.

### C. 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 \( \Delta_{m\to nl}(t, \cdot) \) can be converted to the pdf \( \frac{1}{C} \exp(\Delta_{m\to nl}^x(t, \cdot)) \), where the choice of scale factor \( C = \int_{x_{nl}} \exp(\Delta_{m\to nl}^x(t, x_{nl})) \) ensures that the pdf integrates to one. Four types of message will be used, as specified in Table I. We also find it convenient to express the (iteration-\( t \) SPA-approximated) posterior pdfs \( p_{nk\,l}(Y \mid t, l) \) and \( p_{mn\,l}(Y \mid t, n) \) in the log domain as \( \Delta_{m\to nl}(t, \cdot) \) and \( \Delta_{n\to ml}(t, \cdot) \), respectively, again with arbitrary constant offsets.

Applying the SPA to the factor graph in Fig. 1, we arrive at the following update rules for the four messages in Table I

\[
\Delta_{m\to nl}(t, x_{nl}) = \log \prod_{r \neq n} \left( p_{ml\,r}(z_{ml} \mid y_{ml}) \prod_{k=1}^{N} a_{mk} x_{kl} \right) \quad (7)
\]

\[
\Delta_{n\to ml}(t, a_{mn}) = \log \prod_{r \neq m} \left( p_{ml\,r}(z_{ml} \mid y_{ml}) \prod_{k=1}^{N} a_{mk} x_{kl} \right) \quad (8)
\]

\[
\Delta_{k\to mn}(t, a_{mn}) = \log \prod_{r \neq k} \left( p_{ml\,r}(z_{ml} \mid y_{ml}) \prod_{k=1}^{N} a_{mk} x_{kl} \right) \quad (9)
\]

where \( \text{const} \) is an arbitrary constant (w.r.t \( x_{nl} \) in (7) and (8), and w.r.t \( a_{mn} \) in (9) and (10)). In the sequel, we denote the mean and variance of the pdf \( \frac{1}{C} \exp(\Delta_{m\to nl}^x(t, \cdot)) \) by \( \mu_{m\to nl}^x(t) \) and \( \sigma_{m\to nl}^x(t) \), respectively, and we denote the mean and variance of \( \frac{1}{C} \exp(\Delta_{n\to ml}^a(t, \cdot)) \) by \( \mu_{n\to ml}^a(t) \) and \( \sigma_{n\to ml}^a(t) \). For the log-posteriors, the SPA implies

\[
\Delta_{m\to nl}(t+1, x_{nl}) = \log p_{nk\,l}(x_{nl}) + \sum_{m=1}^{M} \Delta_{m\to nl}(t, x_{nl}) + \text{const} \quad (11)
\]

\[
\Delta_{n\to ml}(t+1, a_{mn}) = \log p_{mn\,l}(a_{mn}) + \sum_{k \neq l} \Delta_{k\to mn}(t, a_{mn}) + \text{const} \quad (12)
\]

and we denote the mean and variance of \( \frac{1}{C} \exp(\Delta_{m\to nl}^x(t, \cdot)) \) by \( \mu_{m\to nl}(t) \) and \( \sigma_{m\to nl}(t) \), and the mean and variance of \( \frac{1}{C} \exp(\Delta_{n\to ml}^a(t, \cdot)) \) by \( \mu_{n\to ml}(t) \) and \( \sigma_{n\to ml}(t) \).
We now apply AMP approximations to the SPA updates \((7)-(12)\). 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. \(\mathbb{E}\{z_{ml}^2\}\) and \(\mathbb{E}\{x_{ml}^2\}\) scale as \(O(1)\), i.e., that the magnitudes of these elements do not change as \(N \to \infty\). In this case, the relationship \(z_{ml} = \sum_{n=1}^N a_{mn} x_{nl}\) implies that \(\mathbb{E}\{a_{mn}^2\}\) must scale as \(O(1)/N\). These scalings are assumed to hold for random variables \(z_{ml}, a_{mn},\) and \(x_{nl}\) distributed according to the prior pdfs, according to the pdfs corresponding to the SPA messages \((7)-(10)\), and according to the pdfs corresponding to the SPA posterior approximations \((11)-(12)\). These assumptions lead straightforwardly to the scalings of \(\hat{z}_{ml}(t), \nu_{ml}^\nu(t), \hat{x}_{ml}(t), \nu_{ml}^x(t), \hat{a}_{ml}(t), \nu_{ml}^a(t), \hat{\alpha}_{ml}(t), \nu_{ml}^\alpha(t),\) and \(\nu_{ml}^\nu(t)\) specified in Table 11. Furthermore, because \(\Delta_{x_{m,n} \rightarrow x_{n,l}}(t, \cdot)\) and \(\Delta_{a_{m,n} \rightarrow x_{n,l}}(t, \cdot)\) differ by only one term out of \(M\), it is reasonable to assume \((13)-(14)\) that the corresponding difference in means \(\tilde{x}_{m,n}(t) - \tilde{x}_{n,l}(t)\) and variances \(\nu_{m,n,l}(t) - \nu_{n,l}(t)\) are both \(O(1/\sqrt{N})\), which then implies that \(\tilde{x}_{m,n}(t) - \tilde{x}_{n,l}(t)\) is also \(O(1/\sqrt{N})\). Similarly, because \(\Delta_{a_{m,n} \rightarrow x_{n,l}}(t, \cdot)\) and \(\Delta_{a_{m,n} \rightarrow x_{n,l}}(t, \cdot)\) differ by only one term out of \(N\), where \(\delta_{l,mn}(t)\) and \(\delta_{l,mn}(t)\) are \(O(1/\sqrt{N})\), it is reasonable to assume that \(\tilde{a}_{m,n}(t) - \tilde{a}_{m,n}(t)\) is \(O(1/N)\) and that both \(\nu_{m,n,l}(t) - \nu_{m,n,l}(t)\) and \(\tilde{a}_{m,n}(t) - \tilde{a}_{m,n}(t)\) are \(O(1/N^{3/2})\).

The remaining entries in Table 11 will be explained below.

We start by approximating the message \(\Delta_{x_{m,n} \rightarrow x_{n,l}}(t, \cdot)\). Expanding \((13)\), we find

\[
\Delta_{x_{m,n} \rightarrow x_{n,l}}(t, x_{nl}) = \log \int_{\{a_{mn}\}} \mathbb{P}_{ml}(y_{ml} | a_{mn} x_{nl} + \sum_{k=1}^N a_{mn} x_{kl}) N(y_{ml} | a_{mn} x_{nl} + \sum_{k=1}^N a_{mn} x_{kl}) dx_{nl} \times \prod_{\tau \neq n} \exp \left( \Delta_{a_{m,n} \rightarrow x_{n,l}}(t, x_{nl}) \right) \prod_{k=1}^N \exp \left( \Delta_{a_{m,n} \rightarrow x_{n,l}}(t, a_{mn}) \right) + \text{const.}
\]

(13)

For large \(N\), the CLT motivates the treatment of \(z_{ml}\), the random variable associated with the \(z_{ml}\) identified in \(13\), conditioned on \(x_{nl} = x_{nl}\), as Gaussian and thus completely characterized by a (conditional) mean and variance. Defining the zero-mean r.v.s \(\tilde{a}_{l,ml} \triangleq a_{mn} - \hat{a}_{l,ml}(t)\) and \(\tilde{x}_{m,ml} = x_{nl} - \tilde{x}_{m,ml}(t)\), where \(a_{mn} \sim \frac{1}{\sqrt{N}} \exp(\Delta^a_{a_{m,n} \rightarrow x_{n,l}}(t, \cdot))\) and \(x_{nl} \sim \frac{1}{\sqrt{N}} \exp(\Delta^x_{x_{m,n} \rightarrow x_{n,l}}(t, \cdot))\), we can write

\[
z_{ml} = (\hat{a}_{l,ml}(t) + \hat{a}_{l,ml}(t)) x_{nl} + \sum_{k \neq n} (\hat{a}_{l,ml}(t) \tilde{a}_{m,ml}(t) + \hat{a}_{l,ml}(t) \tilde{a}_{m,ml}(t))
\]

(14)

We note that it is straightforward to see that

\[
E\{z_{ml} | x_{ml} = x_{ml}\} = \hat{a}_{l,ml}(t) x_{nl} + \hat{p}_{nl}(t)
\]

(15)

and

\[
\text{var}\{z_{ml} | x_{ml} = x_{ml}\} = \nu^a_{l,ml}(t) x_{nl}^2 + \nu^p_{nl}(t)
\]

(16)

for

\[
\hat{p}_{nl}(t) \triangleq \sum_{k \neq n} (\hat{a}_{l,ml}(t) \nu^x_{m,ml}(t) + \nu^a_{l,ml}(t)x_{nl}^2)
\]

(17)

\[
\nu^p_{nl}(t) \triangleq \sum_{k \neq n} (\hat{a}_{l,ml}(t) \nu^x_{m,ml}(t) + \nu^a_{l,ml}(t)x_{nl}^2) + \nu^a_{l,ml}(t) x_{nl}^2.
\]

(18)

With this conditional-Gaussian approximation, \(13\) becomes

\[
\Delta_{x_{m,n} \rightarrow x_{n,l}}(t, x_{nl}) \approx \text{const} + \log \int_{y_{ml}} \mathbb{P}_{ml}(z_{ml} | y_{ml} | z_{ml}) \times N(z_{ml} | \hat{a}_{l,ml}(t) x_{nl} + \hat{p}_{nl}(t), t^a_{l,ml}(t) x_{nl}^2 + t^p_{nl}(t))
\]

(19)

\[
= H_{ml}(\hat{a}_{l,ml}(t) x_{nl} + \hat{p}_{nl}(t), t^a_{l,ml}(t) x_{nl}^2 + t^p_{nl}(t)) + \text{const}
\]

(20)

terms of the function

\[
H_{ml}(\hat{q}, \nu^a; y) \triangleq \log \int_{z} \mathbb{P}_{ml}(z_{ml} | y | z) N(z; \hat{q}, \nu^a).
\]

(21)

Unlike the original SPA message \((7)\), the approximation \((20)\) requires only a single integration. Still, additional simplifications are possible. First, notice that \(\hat{p}_{nl}(t)\) and \(\nu^p_{nl}(t)\) differ from the corresponding \(n\)-invariant quantities

\[
\hat{p}_{nl}(t) \triangleq \sum_{k=1}^N \hat{a}_{l,ml}(t) \tilde{a}_{m,ml}(t)
\]

(22)

\[
\nu^p_{nl}(t) \triangleq \sum_{k=1}^N (\hat{a}_{l,ml}(t) \nu^x_{m,ml}(t) + \nu^a_{l,ml}(t) \tilde{a}_{m,ml}(t)^2)
\]

(23)

\[
+ \nu^a_{l,ml}(t) \tilde{a}_{m,ml}(t)^2.
\]
by one term. In the sequel, we will assume that \( \hat{\rho}_{ml}(t) \) and \( \nu^p_{ml}(t) \) are \( O(1) \) since these quantities can be recognized as the mean and variance, respectively, of an estimate of \( \mathbf{Z}_{ml} \), which is \( O(1) \). Writing the \( H_{ml} \) term in (20) using (23),

\[
H_{ml} \left( a_{lm,mn}(t)x_{nl} + \hat{\rho}_{ml}(t), \nu^p_{lm,mn}(t)x_{nl} + \nu^p_{ml}(t), y_{ml} \right)
\]

\[
= H_{ml} \left( a_{lm,mn}(t)(x_{nl} - \hat{x}_{ml}(t)) + \hat{\rho}_{ml}(t), \nu^p_{lm,mn}(t)(x_{nl} - \hat{x}_{ml}(t)) + \nu^p_{ml}(t), \nu^p_{ml}(t) \right) + O(1/N),
\]

\[
\nu^p_{lm,mn}(t) \left( x_{nl} - \hat{x}_{ml}(t) \right) + \nu^p_{ml}(t) \left( x_{nl} - \hat{x}_{ml}(t) \right) + O(1/N),
\]

(24)

where in (25) we used the facts that \( \hat{a}_{lm,mn}(t)(\hat{x}_{nl}(t) - \hat{x}_{ml}(t)) \) and \( \nu^p_{lm,mn}(t)(\hat{x}_{nl}(t) - \hat{x}_{ml}(t)) \) and \( \nu^p_{ml}(t)(\hat{x}_{nl}(t) - \hat{x}_{ml}(t)) \) are both \( O(1/N) \).

Rewriting (20) using a Taylor series expansion in \( x_{nl} \) about the point \( \hat{x}_{ml}(t) \), we get

\[
\Delta^x_{m-nl}(t, x_{nl}) \approx \text{const} + H_{ml} (\hat{\rho}_{ml}(t) + O(1/N), \nu^p_{ml}(t) + O(1/N); y_{ml})
\]

\[
+ \frac{1}{2} \left( \hat{a}_{lm,mn}(t) \right)^2 + H_{ml} (\hat{\rho}_{ml}(t) + O(1/N), \nu^p_{ml}(t) + O(1/N); y_{ml})
\]

\[
+ \nu^p_{lm,mn}(t) \left( x_{nl} - \hat{x}_{ml}(t) \right)^2 + H_{ml} (\hat{\rho}_{ml}(t) + O(1/N), \nu^p_{ml}(t) + O(1/N); y_{ml})
\]

\[
+ \nu^p_{ml}(t) \left( x_{nl} - \hat{x}_{ml}(t) \right)^2 + O(1/N^3/2),
\]

(26)

where \( H_{ml} \) and \( H''_{ml} \) are the first two derivatives of \( H_{ml} \) w.r.t. its first argument and \( H_{ml} \) is the first derivative w.r.t its second argument. Note that, in (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 (26) by dropping terms that vanish, relative to the second-to-last term in (26), as \( N \to \infty \). Since this second-to-last term is \( O(1/N) \) due to the scalings of \( \hat{a}_{lm,mn}(t), \hat{\rho}_{ml}(t), \) 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 \( \nu^p_{lm,mn}(t) \) with \( \nu^p_{ml}(t) \) and \( \hat{a}^2_{lm,mn}(t) \) with \( \hat{a}^2_{ln}(t) \), 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 \( O(1/N) \) 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., [30, 31, and 32].

Applying these approximations to (26) and absorbing \( x_{nl} \)-invariant terms into the \( \text{const} \) term, we obtain

\[
\Delta^x_{m-nl}(t, x_{nl}) \approx \left[ \hat{s}_{ml}(t) \hat{a}_{lm,mn}(t) + \nu^p_{ml}(t) \hat{a}^2_{ln}(t) \hat{x}_{nl}(t) \right] \times x_{nl} - \frac{1}{2} \left[ \nu^p_{ml}(t) \hat{a}^2_{ln}(t) - \nu^p_{ml}(t) \right] \times \left( \hat{s}_{ml}(t) - \nu^p_{ml}(t) \right) x_{nl}^2 + \text{const},
\]

(27)

where we used the relationship

\[
H_{ml}(\hat{q}_t, \nu^p_t; y) = \frac{1}{2} \left[ (H_{ml}(\hat{q}_t, \nu^p_t; y))^2 + H''_{ml}(\hat{q}_t, \nu^p_t; y) \right]
\]

and defined

\[
\hat{s}_{ml}(t) \equiv H_{ml}(\hat{\rho}_{ml}(t), \nu^p_{ml}(t); y_{ml})
\]

\[
\nu^p_{ml}(t) \equiv -H''_{ml}(\hat{\rho}_{ml}(t), \nu^p_{ml}(t); y_{ml}).
\]

Note that (27) is essentially a Gaussian approximation to the pdf \( \frac{1}{\sqrt{2\pi}} \exp(\Delta^x_{m-nl}(t, .)) \).

We show in Appendix A that

\[
\hat{s}_{ml}(t) = \frac{1}{\nu^p_{ml}(t)} (\hat{x}_{ml}(t) - \hat{\rho}_{ml}(t))
\]

\[
\nu^p_{ml}(t) = \frac{1}{\nu^p_{ml}(t)} \left( 1 - \frac{\nu^p_{ml}(t)}{\nu^p_{ml}(t)} \right),
\]

(31)

(32)

for the conditional mean and variance

\[
\hat{x}_{ml}(t) \equiv \mathbb{E} \left[ z_{ml} \left| p_{ml} = \hat{\rho}_{ml}(t); \nu^p_{ml}(t) \right. \right]
\]

\[
\nu^p_{ml}(t) \equiv \text{var} \left[ z_{ml} \left| p_{ml} = \hat{\rho}_{ml}(t); \nu^p_{ml}(t) \right. \right],
\]

(33)

computed according to the (conditional) pdf

\[
p_{z_{ml}|p_{ml}}(z_{ml} | p_{ml}; \nu^p_{ml}(t)) \equiv \mathbb{P}_{Y \mid z} (y_{ml} | z_{ml}) \mathcal{N} \left( z_{ml}; \hat{\rho}_{ml}(t), \nu^p_{ml}(t) \right),
\]

(35)

where here \( C = \int \mathbb{P}_{Y \mid z} (y_{ml} | z) \mathcal{N} \left( z; \hat{\rho}_{ml}(t), \nu^p_{ml}(t) \right) \). In fact, (35) is BiG-AMP’s iteration-\( t \) approximation to the true marginal posterior \( p_{z_{ml}|Y}(\cdot | Y) \). We note that (35) can also be interpreted as the (exact) posterior pdf for \( z_{ml} \) given the likelihood \( \mathbb{P}_{Y \mid z} (y_{ml} | z) \mathcal{N} \left( z; \hat{\rho}_{ml}(t), \nu^p_{ml}(t) \right) \) and the prior \( z_{ml} \sim \mathcal{N}(\hat{\rho}_{ml}(t), \nu^p_{ml}(t)) \) that is implicitly assumed by iteration-\( t \) BiG-AMP. Since \( \bar{X}^T = \mathbf{X}^T \mathbf{A}^T \), the derivation of the BiG-AMP approximation of \( \Delta^a_{m-nl}(t, .) \) closely follows the definition for \( \Delta^x_{m-nl}(t, .) \). In particular, it starts with (similar to (31))

\[
\Delta^a_{m-nl}(t, a_{mn}) = \log \int \mathbb{P}_{a_{mn}|x} (a_{mn} | x_{rl}) \prod_{r=1}^{N} \exp \left( \Delta^a_{m-rl}(t, x_{rl}) \right) \prod_{k \neq n} \exp \left( \Delta^a_{l-k}(t, a_{mn}) \right) + \text{const},
\]

(36)

where again the CLT motivates the treatment of \( z_{nl} \), conditioned on \( a_{mn} \) as \( a_{mn} \), Gaussian. Eventually we arrive at the Taylor-series approximation (similar to (27))

\[
\Delta^a_{m-nl}(t, a_{mn}) \approx \left[ \hat{s}_{ml}(t) \hat{a}_{lm,mn}(t) + \nu^p_{ml}(t) \hat{a}^2_{ln}(t) \hat{x}_{nl}(t) \right] \times a_{mn} - \frac{1}{2} \left[ \nu^p_{ml}(t) \hat{a}^2_{ln}(t) - \nu^p_{ml}(t) \right] \times \left( \hat{s}_{ml}(t) - \nu^p_{ml}(t) \right) a^2_{mn} + \text{const},
\]

(37)
E. Approximated Variable-to-Factor Messages

We now turn to approximating the messages flowing from the variable nodes to the factor nodes. Starting with (8) and plugging in (27) we obtain

\[ \Delta_{m\in\mathbb{N}}^x(t+1, x_{nl}) \]

\[ \approx \text{const} + \log \mathbb{P}_{x_{nl}}(x_{nl}) + \sum_{k \neq m} \left[ \mathbb{S}_{x_{nl}}(t) \mathbb{A}_{k,n}(t) + \nu_{x_{nl}}(t) \mathbb{S}_{k,n}(t) \right] \]

\[ + \nu_{x_{nl}}(t) \mathbb{S}_{x_{nl}}(t) \mathbb{S}_{k,n}(t) - \nu_{x_{nl}}(t) \mathbb{S}_{x_{nl}}(t) \mathbb{S}_{k,n}(t) \]

\[ = \text{const} + \log \mathbb{P}_{x_{nl}}(x_{nl}) - \frac{1}{2 \nu_{x_{nl}}(t)} (x_{nl} - \bar{x}_{m,n}(t))^2 \]

\[ = \text{const} + \log \left( \mathbb{P}_{x_{nl}}(x_{nl}) \mathcal{N}(x_{nl}; \bar{x}_{m,n}(t), \nu_{m,n}(t)) \right) \]

for

\[ \nu_{m,n}^x(t) \approx (\frac{1}{2} \sum_{k \neq m} \mathbb{A}_{k,n}^2(t) \nu_{x_{nl}}(t) \mathbb{S}_{k,n}^2(t) - \nu_{x_{nl}}(t) \mathbb{S}_{k,n}^2(t))^{-1} \]

\[ \bar{x}_{m,n}(t) \approx x_{nl}(t) \left( 1 + \nu_{m,n}^x(t) \sum_{k \neq m} \nu_{x_{nl}}(t) \mathbb{S}_{k,n}^2(t) - \nu_{x_{nl}}(t) \mathbb{S}_{k,n}^2(t) \right) \]

\[ + \nu_{m,n}^x(t) \sum_{k \neq m} \mathbb{A}_{k,n}(t) \mathbb{S}_{k,n} \]

Since \( \mathbb{A}_{m,n}(t) \) and \( \nu_{m,n}(t) \) are \( O(1/N) \), and recalling \( \mathbb{S}_{k,n}^2(t) \) and \( \nu_{x_{nl}}(t) \) are \( O(1) \), we take \( \nu_{m,n}^x(t) \) to be \( O(1) \). Meanwhile, since \( \bar{x}_{m,n}(t) \) is an estimate of \( x_{nl} \), we reason that it is \( O(1) \).

The mean and variance of the pdf associated with the \( \Delta_{m\in\mathbb{N}}^x(t+1,.) \) approximation in (40) are

\[ \bar{x}_{m,n}(t+1) \]

\[ \approx \frac{1}{C} \int_x x \mathbb{P}_{x_{nl}}(x(t)) \mathcal{N}(x(t); \bar{x}_{m,n}(t), \nu_{m,n}(t)) \]

\[ \approx g_{x_{nl}}(\bar{x}_{m,n}(t), \nu_{m,n}(t)) \]

\[ \nu_{m,n}^x(t+1) \]

\[ \approx \frac{1}{C} \int_x |x - \bar{x}_{m,n}(t+1)|^2 \mathbb{P}_{x_{nl}}(x(t)) \mathcal{N}(x(t); \bar{x}_{m,n}(t), \nu_{m,n}(t)) \]

\[ \approx \nu_{m,n}^x(t) g_{x_{nl}}'(\bar{x}_{m,n}(t), \nu_{m,n}(t)) \]

where here \( C = \int_x \mathbb{P}_{x_{nl}}(x(t)) \mathcal{N}(x(t); \bar{x}_{m,n}(t), \nu_{m,n}(t)) \) and \( g_{x_{nl}} \) denotes the derivative of \( g_{x_{nl}} \) with respect to the first argument. The fact that (43) and (44) are related through a derivative was shown in (32).

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

\[ \nu_{m,n}^\nu(t) \approx \left( \sum_{m=1}^{M} \hat{a}_{mn}^2(t) \nu_{mn}(t) - \nu_{mn}(t) \mathbb{S}_{ml}^2(t) \mathbb{S}_{ml}(t) \right)^{-1} \]

\[ \bar{r}_{m,n}(t) \approx \bar{x}_{ml}(t) \left( 1 + \nu_{m,n}^\nu(t) \sum_{m=1}^{M} \nu_{mn}(t) \mathbb{S}_{ml}^2(t) - \nu_{mn}(t) \mathbb{S}_{ml}(t) \right) \]

\[ + \nu_{m,n}^\nu(t) \sum_{m=1}^{M} \hat{a}_{ml}(t) \mathbb{S}_{ml}(t) \]

Comparing (45) to (46) with (41)-(42) and applying previously established scalings from Table II reveals that \( \nu_{m,n}^\nu(t) - \nu_{m,n}^\nu(t) \) is \( O(1/N) \) and that \( \bar{r}_{m,n}(t) = \bar{x}_{ml}(t) - \nu_{m,n}^\nu(t) \hat{a}_{mn}(t) \mathbb{S}_{ml}(t) + O(1/N) \), so that (43) implies

\[ \hat{r}_{m,n}(t+1) \approx g_{x_{nl}}(\bar{r}_{m,n}(t), \nu_{m,n}^\nu(t)) \]

\[ = g_{x_{nl}}(\bar{r}_{m,n}(t) - \nu_{m,n}^\nu(t) \hat{a}_{mn}(t) \mathbb{S}_{ml}(t) + O(1/N), \nu_{m,n}^\nu(t) + O(1/N)) \]

\[ = g_{x_{nl}}(\bar{r}_{m,n}(t) - \nu_{m,n}^\nu(t) \hat{a}_{mn}(t) \mathbb{S}_{ml}(t), \nu_{m,n}^\nu(t) + O(1/N)) \]

\[ = g_{x_{nl}}(\bar{r}_{m,n}(t), \nu_{m,n}^\nu(t)) \]

\[ = g_{x_{nl}}(\bar{r}_{m,n}(t), \nu_{m,n}^\nu(t) + O(1/N)) \]

\[ \approx g_{x_{nl}}(\bar{r}_{m,n}(t), \nu_{m,n}^\nu(t) + O(1/N)) \]

\[ \approx g_{x_{nl}}(\bar{r}_{m,n}(t), \nu_{m,n}^\nu(t) + O(1/N)) \]

\[ \approx x_{nl}(t+1) - \hat{a}_{mn}(t) \mathbb{S}_{ml}(t) \]

Above, (48) follows from taking Taylor series expansions around each of the \( O(1/N) \) perturbations in (47); (49) follows from a Taylor series expansion in the first argument of (48) about the point \( \bar{x}_{m,n}(t) \); and (50) follows by neglecting the \( O(1/N) \) term (which vanishes relative to the others in the large-system limit) and applying the definitions

\[ \bar{x}_{m,n}(t+1) \approx g_{x_{nl}}(\bar{r}_{m,n}(t), \nu_{m,n}^\nu(t)) \]

\[ \nu_{m,n}^\nu(t+1) \approx \nu_{m,n}^\nu(t) g_{x_{nl}}(\bar{r}_{m,n}(t), \nu_{m,n}^\nu(t)) \]

which match (43)-(44) sans the \( m \) dependence. Note that (50) confirms that the difference \( \bar{x}_{m,n}(t) - \nu_{m,n}^\nu(t) \) is \( O(1/\sqrt{N}) \), as was assumed at the start of the BiG-AMP derivation. Likewise, taking Taylor series expansions of \( g_{x_{nl}}^\nu \) in (43) about the point \( \bar{r}_{m,n}(t) \) in the first argument and about the point \( \nu_{m,n}^\nu(t) \) in the second argument and then comparing the result with (52) confirms that \( \nu_{m,n}^\nu(t) - \nu_{m,n}^\nu(t) \) is \( O(1/\sqrt{N}) \).

We then repeat the above procedure to derive an approximation to \( \Delta_{m\in\mathbb{N}}^\nu(t+1,.) \) analogous to (40), whose corresponding mean is then further approximated as

\[ \hat{a}_{m,n}(t+1) \approx \hat{a}_{mn}(t+1) - \bar{x}_{m,n}(t) \mathbb{S}_{ml}(t) \]

for

\[ \hat{a}_{mn}(t+1) \approx g_{x_{nl}}(\bar{q}_{mn}(t), \nu_{mn}^\nu(t)) \]

\[ \nu_{mn}(t+1) \approx \nu_{mn}(t) \| g_{x_{nl}}(\bar{q}_{mn}(t), \nu_{mn}^\nu(t)) \]

\[ g_{x_{nl}}(\bar{q}_{m}, \nu_{m}^\nu) \approx \int_{q} a \mathbb{P}_{x_{nl}}(a) \mathcal{N}(a; \bar{q}_{m}, \nu_{m}^\nu) \]

(56)
where
\[ \nu_{mn}^\varphi(t) = \left( \sum_{l=1}^{L} \hat{s}_{nl}^2(t) \nu_{nl}(t) + \nu_{nl}(t) (\hat{s}_{nl}^2(t) - \nu_{nl}(t)) \right)^{-1}. \]  
(57)

\[ \hat{q}_{mn}(t) \triangleq \tilde{a}_{mn}(t) \left( 1 + \nu_{mn}^\varphi(t) \sum_{l=1}^{L} \nu_{nl}(t) [\hat{s}_{nl}^2(t) - \nu_{nl}(t)] \right) + \nu_{mn}^\varphi(t) \sum_{l=1}^{L} \hat{x}_{m,nl}(t) \hat{s}_{ml}(t). \]  
(58)

Arguments analogous to the discussion following (42) justify the remaining scalings in Table II.

F. Closing the Loop

The penultimate step in the derivation of BiG-AMP is to approximate earlier steps that use \( \hat{a}_{l,mn}(t) \) and \( \tilde{x}_{m,nl}(t) \) in place of \( \tilde{a}_{mn}(t) \) and \( \tilde{x}_{nl}(t) \). For this, we start by plugging (50) and (53) into (22), which yields
\[ \Delta_{\mu}(t+1, x_{nl}) \approx \text{const} + \log \left( p_{x_{nl}}(x_{nl} | \tilde{r}_{nl}(t), \nu_{nl}^\varphi(t)) \right) \]  
(64)

and (57) into (23), giving
\[ \nu_{l,mn}(t) = \nu_{mn}(t) + O(1/N^{3/2}). \]  
(65)

\[ \nu_{mn}^\varphi(t) \equiv \left( \sum_{l=1}^{L} \hat{s}_{nl}^2(t) \nu_{nl}(t) + \nu_{nl}(t) (\hat{s}_{nl}^2(t) - \nu_{nl}(t)) \right)^{-1}. \]  
(66)

where (62) retains only the \( O(1) \) terms from (61). Similarly, we plug (53) into (46) and (50) into (58) to obtain
\[ \nu_{mn}^\varphi(t) \equiv \left( \sum_{l=1}^{L} \hat{s}_{nl}^2(t) \nu_{nl}(t) + \nu_{nl}(t) (\hat{s}_{nl}^2(t) - \nu_{nl}(t)) \right)^{-1}. \]  
(67)

\[ \Delta_{\mu}(t+1, a_{mn}) \approx \text{const} + \log \left( p_{a_{mn}}(a_{mn} | \hat{q}_{mn}(t), \nu_{mn}^\varphi(t)) \right) \]  
(68)

using steps similar to (40). The associated pdfs are
\[ p_{x_{nl}}(x_{nl} | \tilde{r}_{nl}(t), \nu_{nl}^\varphi(t)) \]  
(69)

\[ p_{a_{mn}}(a_{mn} | \hat{q}_{mn}(t), \nu_{mn}^\varphi(t)) \]  
(70)

for \( C_x \) and \( C_a \), which are iteration-t BiG-AMP’s approximations to the true marginal posteriors \( p_{x_{nl}}(x_{nl} | Y) \) and \( p_{a_{mn}}(a_{mn} | Y) \), respectively.

Note that \( \tilde{x}_{nl}(t+1) \) and \( \nu_{mn}^\varphi(t+1) \) from (51) and (52) are the mean and variance, respectively, of the posterior pdf in (69). Note also that (69) can be interpreted as the (exact) posterior pdf of \( x_{nl} \) given the observation \( r_{nl} = \hat{r}_{nl}(t) \) under the prior model \( x_{nl} \sim p_{x_{nl}} \) and the likelihood model \( p_{x_{nl} | x_{nl}}(r_{nl}(t) | x_{nl}; \nu_{nl}^\varphi(t)) = N(r_{nl}(t); x_{nl}, \nu_{nl}^\varphi(t)) \) implicitly assumed by iteration-t BiG-AMP. Analogous statements can be made about the posterior pdf of \( a_{mn} \) in (70).

This completes the derivation of BiG-AMP.
the scaled residual \( \hat{\mathbf{S}} \) and a set of inverse-residual-variables \( \{ \nu^r_{ml} \} \). This interpretation becomes clear in the case of AWGN observations with noise variance \( \nu^w \), where

\[
p^m_{nl}(z_{nl} | y_{ml} ; \nu^w) = \mathcal{N}(y_{ml} ; z_{nl}, \nu^w).
\]

and hence

\[
\nu^r_{ml} = \frac{1}{\nu^p_{ml} + \nu^w} \quad \text{and} \quad \hat{\mathbf{s}}_{ml} = \frac{y_{ml} - \hat{p}_{nl}}{\nu^p_{ml} + \nu^w}.
\]

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

The BiG-AMP algorithm in Table III is a direct (although non-trivial) extension of the GAMP algorithm for compressive sensing [32], which estimates \( \hat{\mathbf{x}} \) assuming perfectly known \( \hat{\mathbf{A}} \), and even stronger similarities to the \( \hat{\mathbf{A}} \)-uncertain GAMP from [49], which estimates \( \hat{\mathbf{x}} \) assuming knowledge of the marginal means and variances of unknown random \( \hat{\mathbf{A}} \), but which makes no attempt to estimate \( \hat{\mathbf{A}} \) itself. In Sec. III-B, a simplified version of BiG-AMP will be developed that is similar to the Bayesian-AMP algorithm [50] for compressive sensing.

### III. BiG-AMP SIMPLIFICATIONS

We now describe simplifications of the BiG-AMP algorithm from Table III that result from additional approximations and from the use of specific priors \( p_{ml} \), \( p_{nl} \), and \( p_{pm,n} \) that arise in practical applications of interest.

#### A. Scalar Variances

The BiG-AMP algorithm in Table III stores and processes a number of element-wise variance terms whose values vary across the elements (e.g., \( \nu^r_{ml} \) can vary across \( m, n \) 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 approximate \( \forall m, n : \nu^r_{ml}(t) \approx \nu^r \) and \( \forall m, n : \nu^r_{mn}(t) \approx \nu^r \). From (R1)

\[
\nu^p_{ml}(t) \approx \nu^p(t) \quad \text{and} \quad \nu^p_{mn}(t) \approx \nu^p(t) \quad \text{so that (R1)}
\]

\[
\nu^p_{ml}(t) \approx \nu^p(t) \quad \text{and} \quad \nu^p_{mn}(t) \approx \nu^p(t) \quad \text{so that (R1)}
\]

\[
\frac{\| \hat{\mathbf{A}}(t) \|^2}{M} \nu^p(t) + \frac{\| \hat{\mathbf{X}}(t) \|^2}{L} \nu^p(t) \approx \nu^p(t).
\]

Note that using (74) in place of (R1) avoids two matrix multiplications. Plugging these approximations into (R3) gives

\[
\nu^p_{ml}(t) \approx \nu^p_{mn}(t) \nu^p(t) \quad \text{and} \quad \nu^p_{mn}(t) \approx \nu^p(t)
\]

which, when used in place of (R3), avoids another matrix multiply. Even with the above scalar-variance approximations, \( \{ \nu^r_{ml} \} \) from (R5) are not guaranteed to be equal (except in special cases like AWGN \( p_{ml} \)). Still, they can be
approximated as such using \( \nu^a(t) \leq \frac{1}{MT} \sum_{m=1}^{M} \sum_{l=1}^{L} \nu_{ml}^a(t) \), in which case
\[
\nu_{ml}^q(t) \approx \frac{1}{\nu^a(t) \sum_{m=1}^{M} |\hat{a}_{mn}(t)|^2} \approx \frac{N}{\nu^a(t) \|A(t)\|^2_F} \approx \nu^a(t).
\]
(76)

Using (76) in place of (R9) and (77) in place of (R11) avoids two matrix multiplies and \( NL + MN - 2 \) scalar divisions, and furthermore allows (R10) and (R12) to be implemented as
\[
\tilde{x}_{nl}(t) = \tilde{x}_{nl}(t) \left( 1 - \frac{MN\nu^a(t)}{\|A(t)\|^2_F} \right) + \nu^a(t) \sum_{m=1}^{M} \tilde{a}_{mn}(t) \hat{s}_{ml}(t)
\]
(78)

\[
\hat{q}_{mn}(t) = \hat{a}_{mn}(t) \left( 1 - \frac{NL\nu^a(t)}{\|X(t)\|^2_F} \right) + \nu^a(t) \sum_{l=1}^{L} \tilde{x}_{nl}(t) \hat{s}_{ml}(t),
\]
(79)
saving two more matrix multiplies, and leaving a total of only three matrix multiplies per iteration.

B. Possibly Incomplete AWGN Observations

We now consider a particular observation model wherein the elements of \( Z = AX \) are AWGN-corrupted at a subset of indices \( \Omega \subset \{1, \ldots M \} \times \{1, \ldots L \} \) and unobserved at the remaining indices, noting that the standard AWGN model (71) 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
\[
P_{Y_{m|l} \mid Z_{ml}}(y_{ml} \mid z_{ml}) = \mathcal{N}(y_{ml}; z_{ml}, \nu^w) \quad (m,l) \in \Omega \]
\[
\mathcal{N}(y_{ml}; \nu_{ml}^w) \quad (m, l) \notin \Omega.
\]
(80)

where \( \nu^w \) is the noise variance on the non-observing indices and \( \nu_{ml}^w \) denotes a point mass at \( y = 0 \). Thus, at the observed entries \( (m,l) \in \Omega \), the quantities \( \hat{s}_{ml} \) and \( \nu_{ml}^a \) calculated using the AWGN expressions (72), while at the “missing” entries \( (m,l) \notin \Omega \), where \( y_{ml} \) is invariant to \( z_{ml} \), we have \( E\{z_{ml} \mid p_{ml} = \hat{p}_{ml} \} = \hat{p}_{ml} \) and \( \text{var}\{z_{ml} \mid p_{ml} = \hat{p}_{ml} \} = \nu_{ml}^w \), so that \( \hat{s}_{ml} = 0 \) and \( \nu_{ml}^a = 0 \). This is expected, given that \( \nu^a(t) \) can be interpreted as an inverse residual variance and \( \hat{s} \) as a \( \nu^a \)-scaled residual. In summary, the PIAWGN model yields
\[
\hat{s}_{ml}(t) = \begin{cases} \frac{y_{ml} - \hat{p}_{ml}(t)}{\nu_{ml}^w(t) + \nu^w} & (m,l) \notin \Omega \\ 0 & (m,l) \in \Omega \end{cases}
\]
(81)

\[
\nu_{ml}^a(t) = \begin{cases} \frac{1}{\nu_{ml}^w(t) + \nu^w} & (m,l) \notin \Omega \\ 0 & (m,l) \in \Omega \end{cases}
\]
(82)

When the PIAWGN model is combined with the scalar-variance approximations from Sec. [II-A] BiG-AMP simplifies considerably. To see this, we start by using \( \nu^p(t) \) from (75) in place of \( \nu_{ml}^p(t) \) in (81)-(82), resulting in
\[
\hat{S}(t) = P_{\Omega} \left( \frac{Y - \hat{P}(t)}{\nu^p(t) + \nu^w} \right)
\]
(83)

\[
\nu^a(t) = \frac{\nu^w + \nu^p(t)}{\delta}
\]
(84)

where \( \delta \triangleq \frac{\Omega}{MT} \) 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
\[
[P_{\Omega}(Z)]_{ml} \triangleq \begin{cases} z_{ml} & (m,l) \in \Omega \\ 0 & (m,l) \notin \Omega \end{cases}.
\]
(85)

We can then write (R10) and (R12) as
\[
\hat{R}(t) = \hat{X}(t) \left( 1 - \frac{MN\nu^a(t)}{\|A(t)\|^2_F} \right) + \frac{N}{\delta\|A(t)\|^2_F} \hat{A}(t) \hat{V}(t)
\]
(86)

\[
\hat{Q}(t) = \hat{A}(t) \left( 1 - \frac{NL\nu^a(t)}{\|X(t)\|^2_F} \right) + \frac{N}{\delta\|X(t)\|^2_F} \hat{V}(t) \hat{X}(t)
\]
(87)

using (78)-(79) and (83)-(85) with
\[
\hat{V}(t) \triangleq P_{\Omega}(Y - \hat{P}(t))
\]
(88)

\[
= P_{\Omega}(Y - \hat{P}(t)) + \nu^p(t) \hat{S}(t-1)
\]
(89)

\[
= P_{\Omega}(Y - \hat{P}(t)) + \frac{\nu^p(t)}{\nu^p(t-1) + \nu^w} \hat{V}(t-1),
\]
(90)

since \( P_{\Omega} \) is a projection operator, and using (R4) and (84).

Scalar-variance BiG-AMP under PIAWGN observations is summarized in Table [LV]. Note that the residual matrix \( \tilde{U}(t) \triangleq P_{\Omega}(Y - \hat{A}(t)\hat{X}(t)) \) needs to be computed and stored only at the observed entries \( (m,l) \in \Omega \), leading to significant savings when the observations are highly incomplete (i.e., \( |\Omega| \ll ML \)). The same is true for the Onsager-corrected residual, \( \hat{V}(t) \). Thus, the algorithm in Table [LV] 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 (41) that bears close similarity 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 Sec. [II] due to the many simplifications offered by this special case.

C. Zero-mean iid Gaussian Priors on A and X

In this section we will investigate the simplifications that result in the case that both \( p_{a,m} \) and \( p_{s,ml} \) are zero-mean iid

\(^5\)Similar computational savings also occur with incomplete non-Gaussian observations.

\(^6\)The approach in [41] does not compute (or use) \( \nu^p(t) \) as given in lines (R4p)-(R5p) of Table [LV] 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:
\[ \tilde{V}(0) = 0 \]
for \( t = 1, \ldots, T_{\text{max}} \)
\[ G_d(t) = \frac{N}{\mathbf{a}(t) \mathbf{b}(t) + \nu_d(t)} \]
\[ G_q(t) = \frac{N}{\mathbf{q}(t) \mathbf{q}(t) + \nu_q(t)} \]
\[ \tilde{U}(t) = P_{\mathbf{y}} \left( \mathbf{y} - \tilde{A}(t) \mathbf{x}(t) \right) \]
\[ \nu_d(t) = \nu_d(t-1) + \nu_d(t) + \nu_d(t) \]
\[ \nu_q(t) = \nu_q(t) + \nu_q(t) \]
\[ \tilde{V}(t) = \tilde{U}(t) + \nu_q(t) \tilde{V}(t-1) \]

TABLE IV
SCALAR-VARIANCE BiG-AMP WITH PIAWGN \( p_i \)

Gaussian, i.e.,
\[ p_{x_n}(x) = \mathcal{N}(x; 0, \nu_d) \quad \forall n, l \]
\[ p_{y_{nl}}(x) = \mathcal{N}(x; 0, \nu_q) \quad \forall m, n, \]
which, as will be discussed later, is appropriate for matrix completion. In this case, straightforward calculations reveal that \( E \left[ x_{n,l} | r_{n,l}, \nu_{n,l} \right] = \bar{r}_{n,l} \nu_{n,l} / (\nu_{n,l} + \nu_q) \) and \( \text{var} (x_{n,l} | r_{n,l}, \nu_{n,l}) = \nu_{n,l}^2 / (\nu_{n,l} + \nu_q) \). Similarly, \( E \left[ y_{m,n} | q_{m,n}, \nu_{m,n} \right] = \bar{q}_{m,n} \nu_{m,n} / (\nu_{m,n} + \nu_q) \) and \( \text{var} (y_{m,n} | q_{m,n}, \nu_{m,n}) = \nu_{m,n}^2 / (\nu_{m,n} + \nu_q) \).

IV. ADAPTIVE DAMPING

The approximations made in the BiG-AMP derivation presented in Sec. III were well-justified in the large system limit, i.e., the case where \( M, N, L \to \infty \) with fixed \( N/M \) and \( L/N \). In practical applications, however, these dimensions (especially \( N \)) are finite, and hence the algorithm presented in Sec. III may diverge. In case of compressive sensing, the use of “damping” with GAMP yields provable convergence guarantees with arbitrary matrices [53]. 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 [50] and coded in [51].

A. Damping

In BiG-AMP, the iteration-\( t \) damping factor \( \beta(t) \in (0, 1) \) is used to slow the evolution of certain variables, namely \( \nu_m, \nu_{mn}, \nu_{ml}, \bar{s}_{ml}, \bar{y}_{nl}, \) and \( \bar{a}_{mn} \). To do this, steps (R1), (R3), (R7), and (R8) in Table III are replaced with

\[ \nu_{ml}(t) = \beta(t) \left( \sum_{n=1}^{N} \tilde{a}_{mn}(t) |\nu_{ml}(t) + \nu_{mn}(t)| \tilde{a}_{mn}(t) \right) \]
\[ + (1 - \beta(t))(\nu_{ml}(t-1)) \]
\[ \nu_{ml}(t) = \beta(t) \left( \nu_{ml}(t) + \sum_{n=1}^{N} \tilde{a}_{mn}(t) \nu_{ml}(t) \right) \]
\[ + (1 - \beta(t))(\nu_{ml}(t-1)) \]
\[ \bar{s}_{ml}(t) = \beta(t) \left( \tilde{s}_{ml}(t - 1) / \nu_{ml}(t) \right) \]
\[ + (1 - \beta(t)) \tilde{s}_{ml}(t-1) \]

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

\[ \nu_{ml}(t) = \bar{s}_{ml}(t) / \nu_{ml}(t) \]
\[ \bar{s}_{ml}(t) = \bar{s}_{ml}(t) / \nu_{ml}(t) \]

B. 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 relative to \( \{ J(t) \}_{T^{-1}=1}^{T} \) 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 [52]. We now describe how the cost criterion \( J(t) \) is constructed, building on ideas in [53].

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

\[
p_{X,A|Y} = \arg \min_{b_{X,A}} D(b_{X,A} \| p_{X,A|Y}). \tag{99}
\]

The factorized form (5) of the posterior allows us to write

\[
D(b_{X,A} \| p_{X,A|Y}) = \log p_Y(Y) - \int_{A,X} b_{X,A}(A,X) \log \frac{b_{X,A}(A,X)}{p_{X|Z}(Y|AX)} p_X(X) p_A(A) \tag{100}
\]

\[
= D(b_{X,A} \| p_{X|AX}) - \int_{A,X} b_{X,A}(A,X) \log p_{Y|Z}(Y|AX) \tag{101}
\]

Equations (99) and (101) then imply that

\[
p_{X,A|Y} = \arg \min_{b_{X,A}} J(b_{X,A}) \tag{102}
\]

\[
J(b_{X,A}) = D(b_{X,A} \| p_{X|AX}) - E_{b_{X,A}} \left\{ \log p_{Y|Z}(Y|AX) \right\}. \tag{103}
\]

To judge whether a given time-\( t \) BiG-AMP approximation “\( b_{X,A}(t) \)” of the joint posterior \( p_{X,A|Y} \) is better than the previous approximation \( b_{X,A}(t-1) \), one could in principle plug the posterior approximation expressions (69)–(70) into (103) and then check whether \( J(b_{X,A}(t)) < J(b_{X,A}(t-1)) \). But, since the expectation in (103) is difficult to evaluate, we approximate the cost by using, in place of \( AX \), an independent Gaussian matrix with whose component means and variances are matched to those of \( AX \). Taking the joint BiG-AMP posterior approximation \( b_{X,A}(t) \) to be the product of the marginals from (69)–(70), the resulting component means and variances are

\[
E_{b_{X,A}(t)} \{ AX \}_{ml} = \sum_n E_{b_{X,A}(t)} \{ \theta_{mn} \} E_{b_{X,A}(t)} \{ x_{nl} \} \tag{104}
\]

\[
\hat{a}_{mn}(t) = \sum_n \hat{a}_{mn}(t) = \mathbf{7}_{ml}(t) \tag{105}
\]

\[
\text{var}_{b_{X,A}(t)} \{ AX \}_{ml} = \sum_n \text{var}_{b_{X,A}(t)} \{ x_{nl} \} \tag{106}
\]

In this way, the approximate iteration-\( t \) cost becomes

\[
\hat{J}(t) = \sum_{n,l} \left\{ D \left( p_{x_{nl}|\hat{r}_{nl}(t); \nu_{nl}(t)} \| p_{x_{nl}(t)} \right) \right\} + \sum_{m,n} \left\{ D \left( p_{a_{mn}|q_{mn}(t); \nu_{mn}(t)} \| p_{a_{mn}(t)} \right) \right\} - \sum_{m,l} E_{z_{ml}} \{ N(p_{z_{ml}(t); \nu_{z_{ml}(t)}}) \} \tag{108}
\]

Intuitively, the first term in (108) 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 \( \mathbf{Z} \).

V. PARAMETER TUNING AND RANK SELECTION

A. PARAMETER TUNING VIA EXPECTATION MAXIMIZATION

Recall that BiG-AMP requires the specification of priors \( p_{X|Z} = \prod_{m,n} p_{x_{nl}|z_{ml}}(x_{nl}; \nu_{z_{ml}}) \), \( p_{A|Z} = \prod_{m,n} p_{a_{mn}|z_{ml}}(a_{mn}; \nu_{z_{ml}}) \), and \( p_{Y|Z}(Y|Z) = \prod_{m,l} p_{y_{ml}|z_{ml}}(y_{ml}|z_{ml}) \). 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 \( p_{y_{ml}|z_{ml}}(y_{ml}|z_{ml}) = N(y_{ml}; z_{ml}, \nu_{z_{ml}}) \), but the noise variance \( \nu_{z_{ml}} \) may be unknown. In this section, we outline a methodology that takes a given set of BiG-AMP parameterized priors \( p_{x_{nl}|z_{ml}}(x_{nl}; \theta), p_{a_{mn}|z_{ml}}(a_{mn}; \theta), p_{y_{ml}|z_{ml}}(y_{ml}|z_{ml}) \) and tunes the parameter vector \( \theta \) using an expectation-maximization (EM) [80] based approach, with the goal of maximizing the likelihood, i.e., finding \( \hat{\theta} = \arg \max \theta \log p_Y(Y; \theta) \). The approach presented here can be considered as a generalization of the GAMP-based work [87] to BiG-AMP.

Taking \( X, A, \) and \( Z \) to be the hidden variables, the EM recursion can be written as

\[
\hat{\theta}^{k+1} = \arg \max \theta \left\{ \log p_{X,A,Z,Y}(X, A, Z, Y; \theta) \right\} \tag{109}
\]

\[
\hat{\theta}^{k+1} = \arg \max \theta \left\{ \log p_{X,A,Z,Y}(X, A, Z, Y; \theta) \right\} \tag{109}
\]

where for (109) we used the fact \( p_{X,A,Z,Y}(X, A, Z, Y; \theta) = p_{X}(X; \theta) p_{A}(A; \theta) p_{Y|Z}(Y|Z; \theta) \) and the factorizability of \( p_{X}, p_{A}, \) and \( p_{Y|Z} \). As can be seen from (109), knowledge of the marginal posteriors \( \{ p_{x_{nl}|z_{ml}}(x_{nl}; z_{ml}) \}_{m,n,l} \) is sufficient to compute the EM update. Since the exact marginal posteriors are unknown, we employ BiG-AMP’s approximations from (69), (70), and (55) for approximate EM. In addition, we adopt the “incremental” update strategy from [54], where the maximization over \( \theta \) 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 PIAWGN model \(^9\). Equation (109) suggests

\[
(\nu^w)^{k+1} = \arg\max_{\nu^w} \sum_{(m,l)\in \Omega} \int_{z_{ml}} p_{z_{ml}|y}(z_{ml}|y) \times \log p_{z_{ml}}(y_{ml}, \nu^w),
\]

where the true marginal posterior \( p_{z_{ml}|y}(z_{ml}|y) \) is replaced with the most recent BiG-AMP approximation \( p_{z_{ml}|y}(z_{ml}|y_{ml}; \nu^w_{ml}(T_{\text{max}})) \), where “most recent” is with respect to both EM and BiG-AMP iterations. Zeroing the derivative of the sum in (110) with respect to both EM and BiG-AMP iterations. Zeroing the derivative of the sum in (110) with respect to \( \nu^w \),

\[
(\nu^w)^{k+1} = \frac{1}{|\Omega|} \sum_{(m,l)\in \Omega} (y_{ml} - \hat{z}_{ml}(T_{\text{max}}))^2 + \nu^2_{ml}(T_{\text{max}}),
\]

(111)

where \( \hat{z}_{ml}(t) \) and \( \nu^w_{ml}(t) \) are the BiG-AMP approximated posterior mean and variance from \(^{33},^{34}\).

The overall procedure can be summarized as follows. From a suitable initialization \( \hat{\theta}^0 \), BiG-AMP is run using the priors \( \{ p_{X_{nl}}(\cdot; \hat{\theta}^0), p_{a_{mn}}(\cdot; \hat{\theta}^0), p_{z_{ml}|y}(y_{ml}; \hat{\theta}^0) \} \) \( \forall m,n,l \) and iterated to completion, yielding approximate marginal posteriors on \( \{ x_{nl}, a_{mn}, z_{ml} \} \) \( \forall m,n,l \). These posteriors are used in (109) to update the parameters \( \theta \) one element at a time, yielding \( \hat{\theta} \). BiG-AMP is then run using the priors \( \{ p_{X_{nl}}(\cdot; \hat{\theta}), p_{a_{mn}}(\cdot; \hat{\theta}), p_{z_{ml}|y}(y_{ml}; \hat{\theta}) \} \) \( \forall m,n,l \), 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 \(^{37}\).

B. 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 \( A \) (and rows in \( X \)) in the matrix factorization \( Z = AX \). 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 \( Y \), building on well-known rank-selection procedures.

1) Penalized log-likelihood maximization: Consider a set of possible models \( \{ \mathcal{H}_N \} \) \( N=1 \) for the observation \( Y \) where, under \( \mathcal{H}_N \), EM-BiG-AMP estimates \( \Theta_N = \{ A_N, X_N, \theta \} \). Here, the subscripts on \( A_N \) and \( X_N \) indicate the restriction to \( N \) columns and rows, \( \theta \) refers to the vector of parameters defined in Sec.\(V.A \) and the subscript on \( \Theta_N \) indicates the dependence of the overall number of parameters in \( \Theta_N \) with the rank \( N \). Because the selection rule \( \hat{N} = \arg \max_N p_Y(\cdot; \mathcal{H}_N) \) is typically intractable, several well-known rules of the form

\[
\hat{N} = \arg \max_{N=1,\ldots,N} 2 \log p_Y(\Theta_N | \Theta_N) - \eta(N)
\]

have been developed, such as the Bayesian Information Criterion (BIC) and Akaike’s Information Criterion (AIC) \(^{55}\). In \(^{12}\), \( \Theta_N \) is the ML estimate of \( \Theta_N \) under \( Y \), 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 \( Y \).

Applying this methodology to EM-BiG-AMP, where \( p_Y(Y | \Theta_N) = p_Y(Z | A_NX_N; \theta) \), we obtain the rank-selection rule

\[
\hat{N} = \arg \max_{N=1,\ldots,N} 2 \log p_Y(Y | A_N \hat{X}_N; \hat{\theta}) - \eta(N).
\]

(113)

Since \( N_{\text{eff}} \) depends on the application (e.g., matrix completion, robust PCA, dictionary learning), detailed descriptions of \( \eta(\cdot) \) are postponed to \(^{56}\).

To perform the maximization over \( N \) in (113), we start with a small hypothesis \( N_1 \) and run EM-BiG-AMP to completion, generating the (approximate) MMSE estimates \( \hat{A}_{N_1}, \hat{X}_{N_1}, \) and ML estimate \( \hat{\theta} \), which are then used to evaluate \(^8\) the penalized log-likelihood in (113). The \( N \) hypothesis is then increased by a fixed value (i.e., \( N_2 = N_1 + \tau \text{ rankStep} \)), initializations of \( (A_{N_2}, X_{N_2}, \theta) \) are chosen based on the previously computed \( (A_{N_1}, X_{N_1}, \theta) \), and EM-BiG-AMP is run to completion, yielding estimates \( (\hat{A}_{N_2}, \hat{X}_{N_2}, \hat{\theta}) \) with which the penalized likelihood is again evaluated. This process continues until either the value of the penalized log-likelihood decreases, in which case \( \hat{N} \) is set to the previous (i.e., maximizing) hypothesis of \( N \), or the maximum-allowed rank \( \bar{N} \) is reached.

2) Rank contraction: We now describe an alternative rank-selection procedure that is appropriate when \( Z \) has a “cliff” in its singular value profile and which is reminiscent of that used in LMaFit \(^{12}\). In this approach, EM-BiG-AMP is initially configured to use the maximum-allowed rank, i.e., \( N = \bar{N} \). After the first EM iteration, the singular values \{\(\sigma_n\)\} of the estimate \( \hat{X} \) and the corresponding pairwise ratios \( R_n = \sigma_n / \sigma_{n+1} \) are computed \(^{10}\) from which a candidate rank estimate \( \hat{N} = \arg \max_n R_n \) 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

\[
R_{\hat{N}} > \frac{\tau_{\text{MOS}}}{N - 2} \sum_{i \neq \hat{N}} R_i,
\]

(114)

and if \( \hat{N} / \bar{N} \) 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 \( \hat{N} \) is accepted, then the matrices \( A \) and \( X \) are pruned to size \( \hat{N} \) and EM-BiG-AMP is run to convergence. If not, EM-BiG-AMP is run for one more iteration, after which a new candidate \( \hat{N} \) 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 \( Z \) decay smoothly, this procedure can mis-estimate the rank, as discussed in \(^{12}\).

\(^8\)Since we compute approximate MMSE estimates rather than ML estimates, we are in fact evaluating a lower bound on the penalized log-likelihood.

\(^{10}\)In some cases the singular values of \( A \) could be used instead.
VI. 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 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 paper, 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 [https://sourceforge.net/projects/gampmatlab/] including examples of BiG-AMP applied to the MC, RPCA, and DL problems.

A. Problem setup

In matrix completion (MC) [82], one seeks to recover a rank-$N \ll \min(M, L)$ matrix $Z \in \mathbb{R}^{M \times L}$ after observing a fraction $\delta = \frac{|I|}{MN}$ of its (possibly noise-corrupted) entries, where $\Omega$ denotes the set of observations.

BiG-AMP approaches the MC problem by modeling the complete matrix $Z$ as the product $Z = AX$ of random matrices $A \in \mathbb{R}^{M \times N}$ and $X \in \mathbb{R}^{N \times L}$ with priors of the decoupled form in (114-115), where $Z$ is probabilistically related to the observed matrix $Y$ through a likelihood $p_{Y|Z}(Y|Z)$ of the decoupled form in (3). To finally perform MC, BiG-AMP infers $A$ and $X$ from $Y$ under the above model. The corresponding estimates $\hat{A}$ and $\hat{X}$ can then be multiplied to yield an estimate $\hat{Z} = \hat{A}\hat{X}$ of the noiseless complete matrix $Z$.

As in several existing Bayesian approaches to matrix completion (e.g., [23, 53-60]), we choose Gaussian priors for the factors $A$ and $X$. Although EM-BiG-AMP readily supports the use of priors with row- and/or column-dependent parameters, we focus on iid settings of the form

$$p_{A_{mn}}(a) = \mathcal{N}(a; 0, 1) \forall m, n,$$  \hspace{1cm} (115)

$$p_{X_{nl}}(x) = \mathcal{N}(x; \tilde{x}_{nl}, \nu_0^2) \forall n, l,$$ \hspace{1cm} (116)

where the mean and variance in (116) can be tuned using EM-BiG-AMP, as described in the sequel, and where the variance in (115) is fixed to avoid a scaling ambiguity between $A$ and $X$. Section VI-C 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 (80) for the likelihood $p_{Y|Z}$ given by

$$p_{Y_{ml}}(y_{ml} | z_{ml}) = \mathcal{N}(y_{ml}; z_{ml}, \nu^2) \quad (m, l) \in \Omega,$$ \hspace{1cm} (117)

$$p_{Y_{ml}}(y_{ml} | z_{ml}) = \mathcal{N}(y_{ml}; z_{ml}, \nu^2) \quad (m, l) \notin \Omega.$$

Note that, by using (115)-116 with $\tilde{x}_0 = 0$ and the scalar-variance approximation from Sec. II-A, the BiG-AMP algorithm from Table III reduces to the simpler BiG-AMP-Lite algorithm from Table IV with $\nu_0 = 1$.

B. Initialization

In most cases we advocate initializing the BiG-AMP quantities $X(1)$ and $A(1)$ using random draws from the priors $p_X$ and $p_A$, although setting either $X(1)$ or $A(1)$ at zero also seems to perform well in the MC application. Although it is also possible to use SVD-based initializations of $\tilde{X}(1)$ and $\hat{A}(1)$ (i.e., for SVD $Y = U \Sigma D^T$, set $A(1) = U \Sigma^{1/2}$ and $X(1) = \Sigma^{1/2} D^T$) as done in LMaFit [12] and VSBL [23], experiments suggest that the extra computation required is rarely worthwhile for BiG-AMP.

As for the initializations $\nu_{mn}^2(1)$ and $\nu_{mn}^2(1)$, we advocate setting them at 10 times the prior variances in (115-116), which has the effect of weighting the measurements $Y$ more than the priors $p_X, p_A$ during the first few iterations.

C. Adaptive damping

For the assumed likelihood (117) and priors (115)-116, the adaptive-damping cost criterion $\bar{J}(t)$ described in Sec. IV-B reduces to

$$\bar{J}(t) = \frac{1}{2} \sum_{n, l} \left[ \log \frac{\nu_0^2}{\nu_{nl}(t)} + \left( \frac{\nu_{nl}(t)}{\nu_0^2} - 1 \right) + \frac{(\tilde{x}_{nl}(t) - \tilde{x}_0)^2}{\nu_{nl}(t)} \right]$$

$$+ \frac{1}{2} \sum_{m, n} \left[ \log \frac{1}{\nu_{mn}(t)} + \left( \nu_{mn}(t) - 1 \right) + \sigma_{mmn}^2(t) \right]$$

$$+ \frac{1}{\nu^2} \sum_{(m, l) \in \Omega} (y_{ml} - \mathbb{E}(Z_{ml}))^2 + \nu^2(t)$$

$$+ \| \Omega \| \log 2\pi \nu^2.$$ \hspace{1cm} (118)

To derive (118), one can start with the first term in (108) and leverage the Gaussianity of the approximated posterior on $x_{nl}$:

$$\sum_{n, l} \left\| \mathbb{E}(x_{nl} | \tilde{r}_{nl}(t), \nu_{nl}(t)) \right\| p_{x_{nl}}(\cdot)$$ \hspace{1cm} (119)

$$\approx \sum_{n, l} \int_{x_{nl}} \mathcal{N}(x_{nl}; \tilde{x}_{nl}(t), \nu_{nl}(t)) \log \frac{\mathcal{N}(x_{nl}; \tilde{x}_{nl}(t), \nu_{nl}(t))}{\mathcal{N}(x_{nl}; \tilde{x}_0, \nu_0^2)} dx_{nl},$$

which then directly yields the first term in (113). The second term in (118) follows using a similar procedure, and the third and fourth terms follow directly from the PIAWGN model.

In the noise free setting (i.e., $\nu^2 \to 0$), the third term in (118) dominates, omitting the need to compute the other terms.
D. EM-BiG-AMP

For the likelihood (117) and priors (115)-(116), the distributional parameters \( \theta = [p^u, \bar{x}_0, v^0] \) can be tuned using the EM approach from Sec. V-A. To initialize \( \theta \) for EM-BiG-AMP, we adopt the procedure outlined in (17) to our matrix-completion problem, giving the EM initializations \( \bar{x}_0 = 0 \) and

\[
\nu^w = \frac{\| P_{\Omega}(\mathbf{Y}) \|_F^2}{(SNR^0 + 1)|\Omega|}, \quad \nu^s_0 = \frac{1}{N} \left( \frac{\| P_{\Omega}(\mathbf{Y}) \|_F^2}{|\Omega|} - \nu^w \right),
\]

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

E. Rank selection

For MC rank-selection under the penalized log-likelihood strategy (113), we recommend using the small corrected AIC (AIcC) [55] penalty \( \eta(N) = 2 \frac{|\Omega|}{|\Omega| - N_{\text{eff}} - 1} \). For the MC problem, \( N_{\text{eff}} = df + 3 \), where \( df = N(M + L - N) \) counts the degrees-of-freedom in a rank-\( N \) real-valued \( M \times L \) matrix [57] and the three additional parameters come from \( \theta \). Based on the BIAGWN likelihood (117) and the standard form of the ML estimate of \( \nu^w \) (see, e.g., [55 eq. (7)]), the update rule (113) becomes

\[
\hat{N} = \arg \max_{N=1,\ldots,N_{\text{max}}} \left[ \log \left( \frac{1}{|\Omega|} \sum_{(m,l) \in \Omega} (y_{ml} - \bar{x}_{ml}(t))^2 \right) - 2 \frac{|\Omega|}{|\Omega| - N(M + L - N) + 3} \right] - \frac{|\Omega|}{|\Omega| - N(M + L - N) - 4}.
\]

We note that a similar rule (but based on BIC rather than AIcC) was used for rank-selection in [14].

MC rank selection can also be performed using the rank contraction scheme described in Sec. V-B2. We recommend choosing the maximum rank \( N \) to be the largest value such that \( N(M + L - N) < |\Omega| \) and setting \( \tau_{\text{MOS}} = 1.5 \). Since the first EM iteration runs BiG-AMP with the large value \( N = \hat{N} \), we suggest limiting the number of allowed BiG-AMP iterations during this first EM iteration to \( \text{nItFirstEM} = 50 \). 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.

F. 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 [8], a nuclear-norm based convex-optimization method; LMaFit [12], a non-convex optimization-based approach using non-linear successive over-relaxation; GROUSE [16], which performs gradient descent on the Grassmannian manifold; Matrix-ALPS [13], a greedy hard-thresholding approach; and VSBL [23], a variational Bayes approach. In general, we configured BiG-AMP as described in Sec. V-F12 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.

1) Low-rank matrices: We first investigate recovery of rank-\( N \) matrices \( \mathbf{Z} \in \mathbb{R}^{M \times L} \) 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) \( \| \mathbf{Z} - \mathbf{Z}_{\hat{N}} \|^2_F / \| \mathbf{Z} \|^2_F \) of the estimate \( \mathbf{Z} \) returned by the various algorithms under test, examining 10 realizations of \( \{\mathbf{Z}, \Omega\} \) at each problem size \( (M, L, N) \). Here, each realization of \( \mathbf{Z} \) was constructed as \( \mathbf{Z} = \mathbf{A} \mathbf{X} \) for \( \mathbf{A} \) and \( \mathbf{X} \) with iid \( N(0,1) \) elements. All algorithms were forced to use the true rank \( N \), run under default settings with very minor modifications [13] and terminated when the normalized change in either \( \mathbf{Z} \) or \( P_{\Omega}(\mathbf{Z}) \) across iterations fell below the tolerance value of \( 10^{-8} \).

Defining “successful” matrix completion as NMSE < \(-100 \text{ dB}\), Fig. 2 shows the success rate of each algorithm over a grid of sampling ratios \( \delta \in [0, 1] \) and ranks \( N \). As a reference, the solid line superimposed on each subplot delineates the problem feasibility boundary, i.e., the values of \( (\delta, N) \) yielding \( |\Omega| = df \), where \( df = N(M + L - N) \) 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 3 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. 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 Sec. III-A.

Figure 4 plots median runtime [10] to NMSE = \(-100 \text{ dB} \) 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...
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 \( N \).

2) Approximately low-rank matrices: Next we evaluate the performance of recovering approximately low-rank matrices by repeating an experiment from the LMaFit paper \(^{12}\). For this, we constructed the complete matrix as \( Z = U \Sigma V^T \in \mathbb{R}^{500 \times 500} \), where \( U, V \) were orthogonal matrices (built by orthogonalizing iid \( \mathcal{N}(0,1) \) matrices using MATLAB’s \texttt{orth} command) and \( \Sigma \) was a positive diagonal matrix containing the singular values of \( Z \). Two versions of \( \Sigma \) were considered: one with exponentially decaying singular values \( \Sigma_{m,m} = e^{-0.3m} \), and one with the power-law decay \( \Sigma_{m,m} = m^{-3} \).

As in \(^{12}\), we first tried to recover \( Z \) from the noiseless incomplete observations \( \{z_{ml}\}_{(m,l) \in \Omega} \), with \( \Omega \) chosen uniformly at random. Figure 4 shows the performance of several algorithms that are able to learn the underlying rank: LMaFit\(^{13}\), VSBL\(^{14}\) and EM-BiG-AMP under the penalized log-likelihood rank selection strategy from Sec. 3. All three algorithms were allowed a maximum rank of \( N = 30 \). 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.

\(^{17}\)LMaFit was run under the settings provided in their source code for this example.
\(^{18}\)VSBL was allowed at most 100 iterations and run with \texttt{DIMRED\_THR} = 10\(^3\), \texttt{UPDATE\_BETA} = 1, and tolerance = 10\(^{-8}\).
\(^{19}\)Rank-selection rule \(^{113}\) 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\(^{-8}\).

Next, we examined noisy matrix completion by constructing the matrix \( Z = U \Sigma V^T \) as above but then corrupting the measurements with AWGN. Figure 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 \( \delta = 0.2 \). 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.

3) 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 \(^{13}\), where the 512 × 512 boat image was reconstructed from 35% of its pixels sampled uniformly at random. Figure 6 shows the complete (full-rank) image, the images reconstructed by
several matrix-completion algorithms 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 6 also lists the median reconstruction NMSE over 100 realizations, and the NMSE-minimizing rank-40 approximation.

4) Collaborative Filtering: In our final experiment, we investigated the performance of several matrix-completion algorithms on the task of collaborative filtering. For this, we repeated an experiment from the VSBL paper that used the MovieLens 100k dataset, which contains ratings $\{z_{ml}\}_{(m,l) \in \Omega}$, where $|\Omega| = 100\,000$ 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 $\{\hat{z}_{ml}\}_{(m,l) \in \mathcal{R}\setminus\Omega}$.

Performance was then assessed by computing the Normalized Mean Absolute Error (NMAE)

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

(123)

where the 4 in the denominator of (123) 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 7 reports the NMAE and estimated rank $\hat{N}$ for EM-BiG-AMP under the PIAWGN model (117), LMaFit, and VSBL all of which include mechanisms for rank estimation. Figure 7 shows that, under the PIAWGN model, EM-BiG-AMP yields NMAEs that are very close to those of VSBL but slightly inferior at larger training fractions, whereas LMaFit returns NMAEs that are substantially worse at all training fractions. Figure 7 also shows that LMaFit’s estimated rank remains very low (i.e., $\leq 2$) while that of VSBL steadily 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

$$p_{y_{ml}}(y_{ml} | z_{ml}) = \left\{ \begin{array}{ll} \frac{1}{2} \exp \left( -\lambda |y_{ml} - z_{ml}| \right) & (m,l) \in \Omega \\ \frac{1}{2} & (m,l) \notin \Omega, \end{array} \right.$$  

(124)

i.e., a possibly incomplete additive white Laplacian noise (PIAWLN) model, and used the EM-based approach from 21

21VSBL was allowed at most 100 iterations and was run with $\text{DIMRED}_\text{THR} = 10^3$ and $\text{UPDATE}_\text{BETA} = 1$. Both VSBL and EM-BiG-AMP used a tolerance of $10^{-8}$. LMaFit was configured as for the MovieLens experiment in [12]. Each algorithm was allowed a maximum rank of $\hat{N} = 30$.

22The NMAE values reported for VSBL in Fig. 7 are slightly inferior to those reported in [24]. We attribute the discrepancy to differences in experimental setup, such as the construction of $\Omega$.

23The NMAE results presented here differ markedly from those in the MovieLens experiment in [12] 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., $\hat{N} = 94$) yield nearly zero test error.
Sec. V-A to learn the rate parameter $\lambda$. Figure 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 $\hat{N}$ 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.

5) 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).

VII. ROBUST PCA

A. Problem Setup

In robust principal components analysis (RPCA) [61], 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

$$Y = AX + E + W,$$  \hspace{1cm} (125)

where $Z = AX$—the product of tall $A$ and wide $X$—is the low-rank matrix of interest, $E$ is a sparse outlier matrix, and $W$ 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 $A$ as iid $\mathcal{N}(0, \nu^2_A)$ similar to (115), the elements of $X$ as iid $\mathcal{N}(0, \nu^2_X)$ similar to (116), the non-zero
elements of $E$ as iid $\mathcal{N}(0, \nu_1)$, and the elements of $W$ as iid $\mathcal{N}(0, \nu_0)$, with $\nu_1 \gg \nu_0$. In the first approach, $E + W$ is treated as additive noise on $Z$, leading to the likelihood model

$$p_{y_{ml} \mid z_{ml}}(y_{ml} \mid z_{ml}) = (1 - \lambda)\mathcal{N}(y_{ml}; z_{ml}, \nu_0) + \lambda\mathcal{N}(y_{ml}; z_{ml}, \nu_0 + \nu_1),$$

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

In the second approach, $W$ is treated as additive noise but $E$ is treated as an additional estimand. In this case, by multiplying both sides of (125) by any (known) unitary matrix $Q \in \mathbb{R}^{M \times M}$, we can write

$$QY = [QA \ Q] \begin{bmatrix} X \\ E \end{bmatrix} + QW,$$

and apply BiG-AMP to the “augmented” model $Y = AX + W$. Here, $W$ remains iid $\mathcal{N}(0, \nu_0)$, thus giving the likelihood

$$p_{y_{ml} \mid z_{ml}}(y_{ml} \mid z_{ml}) = \mathcal{N}(y_{ml}; z_{ml}, \nu_0).$$

Meanwhile, we choose the following separable priors on $A$ and $X$:

$$p_{\bar{\omega}_{mn}}(\bar{\omega}_{mn}) = \begin{cases} \mathcal{N}(\bar{\omega}_{mn}; 0, \nu_0^0) & n \leq N \\ \mathcal{N}(\bar{\omega}_{mn}; q_{mn}, 0) & n > N \end{cases}$$

and

$$p_{\bar{\omega}_{ml}}(\bar{\omega}_{ml}) = \begin{cases} \mathcal{N}(\bar{\omega}_{ml}; 0, \nu_0^0) & n \leq N \\ (1 - \lambda)\mathcal{N}(\bar{\omega}_{ml}; 0, \nu_1) + \lambda\mathcal{N}(\bar{\omega}_{ml}; 0, \nu_1) & n > N. \end{cases}$$

Essentially, the first $N$ columns of $A$ and first $N$ rows of $X$ model the factors of the low-rank matrix $AX$, and thus their elements are assigned iid Gaussian priors, similar to (115)–(116) in the case of matrix completion. Meanwhile, the last $M$ rows in $X$ are used to represent the sparse outlier matrix $E$, and thus their elements are assigned a Bernoulli-Gaussian prior. Finally, the last $M$ columns of $A$ are used to represent the designed matrix $Q$, and thus their elements are assigned zero-variance priors. Since we find that BiG-AMP is numerically more stable when $Q$ 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 $A$ by left multiplying the estimate of $A$ by $Q^H$.

### B. Initialization

We recommend initializing $\hat{\omega}_{mn}(1)$ using a random draw from its prior and initializing $\hat{\omega}_{ml}(1)$ at the mean of its prior, i.e., $\hat{\omega}_{ml}(1) = 0$. The latter tends to perform better than initializing $\hat{\omega}_{ml}(1)$ randomly, because it allows the measurements $Y$ to determine the initial locations of the outliers in $E$. As in Sec. VI-B, we suggest initializing $\nu_{mml}(1)$ and $\nu_{nml}(1)$ at 10 times the variance of their respective priors to emphasize the role of the measurements during the first few iterations.

### C. EM-BiG-AMP

The EM approach from Sec. V-A can be straightforwardly applied to BiG-AMP for RPCA: after fixing $\nu_0^0 = 1$, EM can be used to tune the remaining distributional parameters, $\theta = [\nu_0, \nu_1, \nu_0^1, \lambda]^T$. To avoid initializing $\nu_0$ and $\nu_0^1$ with overly large values in the presence of large outliers $e_{ml}$, we suggest the following procedure. First, define the set $\Gamma = \{(m, l) : |y_{ml}| \leq \text{median} \{|y_{ml}|\}\}$ and its complement $\Gamma^c$. Then initialize

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

$$\nu_0^1 = \frac{1}{|\Gamma^c|} \sum_{(m, l) \in \Gamma^c} |y_{ml}|^2,$$

where, as in Sec. VI-D, 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.

### D. Rank Selection

In many applications of RPCA, such as video separation, the singular-value profile of $AX$ exhibits a sharp cutoff, in which case it is recommended to perform rank-selection using the contraction strategy from Sec. V-B2.

### E. Avoiding Local Minima

Sometimes, when $N$ is very small, BiG-AMP may converge to a local solution that mistakes entire rows or columns of $AX$ 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 0.8$M$ or any of the column-wise sums exceeds 0.8$L$, 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.
F. Robust PCA Experiments

In this section, we present a numerical study of the two BiG-AMP formulations of RPCA proposed in Sec. VII including a comparison to the state-of-the-art IALM [8], LMaFit [12], GRASTA [19], and VSBL [23] 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.

1) 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 \( Y = Z + E \), where the low-rank component \( Z = AX \) was generated from \( A \) and \( X \) with iid \( N(0, 1) \) entries, and where the sparse corruption matrix \( E \) 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 \( Y \in \mathbb{R}^{M \times N} \) were fixed at \( M = L = 200 \), the rank \( N \) (of \( Z \)) 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 \( Z \), which is the most challenging case: much larger outliers would be easier to detect, after which the corrupted elements of \( Y \) 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. Two versions of IALM were tested: “IALM-1,” which uses the universal penalty parameter \( \lambda_{\text{ALM}} = \frac{1}{\sqrt{M}} \), and “IALM-2,” which tries 50 hypotheses of \( \lambda_{\text{ALM}} \), 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 \( A, X, \) and \( E \) (although their Bernoulli-Gaussian model of \( E \) 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 \( \text{stepMin} = 0.05 \) and \( \text{stepMax} = 0.5 \). Both variants used a maximum of 5 restarts to avoid local minima.

Figure 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 \( Z \). The red curves in Fig. 8 delineate the problem feasibility boundary: for points \((\delta, N)\) above the curve, \( N(M + L - N) \), the degrees-of-freedom in \( Z \), exceeds \((1-\delta)M L \), the number of uncorrupted observations, making it impossible to recover \( Z \) without additional information.

Figure 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 identi-

25We 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.”

26The rank-selection rule (114) was used with \( \text{stepMin} = 5 \), up to 50 EM iterations, and a minimum of 30 and maximum of 500 BiG-AMP iterations per EM iteration.
estimation but rather due to the fact that, at $\delta = 0.1$, these operating points lie above the PTCs shown in Fig. 8.

3) 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 $Y$, run EM-BiG-AMP-2 as described in Sec. VII, extract the background frames from the estimate of the (sparse) outlier component $Z$, and extract the foreground frames from the estimate of the (sparse) outlier component $E$. We note that a perfectly time-invariant background would correspond to a rank-one $Z$ and that the noise term $W$ in (125) can be used to account for small perturbations that are neither low-rank nor sparse.

We tested EM-BiG-AMP$^{27}$ on the popular “mall” video sequence$^{28}$ processing 200 frames (of $256 \times 320$ pixels each) using an initial rank estimate of $\overline{N} = 5$. Figure 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.

VIII. DICTIONARY LEARNING

A. Problem setup

In dictionary learning (DL)$^{62}$, one seeks a dictionary $A \in \mathbb{R}^{M \times N}$ that allows the training samples $Y \in \mathbb{R}^{M \times L}$ to be encoded as $Y = AX + W$ for some sparse coefficient matrix $X$ and small perturbation $W$. 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, $L \gg$
of square dictionaries.

The observations are completely observed.

PIAWGN model (117), but note that in most applications of dependent prior parameters in the case of non-negative matrix considered, such as truncated Gaussian mixtures with column-component variance. However, other priors could be considered, 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. For the likelihood \( p_Y | Z \), we again select the PIAWGN model (117), but note that in most applications of DL the observations are completely observed.

\[ p_{x_{nl}}(x) = (1 - \xi)\delta(x) + \xi N(x; 0, \nu^x), \]  

(134)

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. For the likelihood \( p_Y | Z \), we again select the PIAWGN model (117), but note that in most applications of DL the observations are completely observed.

B. Initialization

In general, we advocate initializing \( \hat{x}_{nl}^{(1)} \) at the mean of the assumed prior on \( x_{nl} \), and initializing the variances \( \nu^x_{nl} \) and \( \nu^z_{mn} \) at 10 times the variance of \( x_{nl} \) and \( a_{mn} \), respectively. We now discuss several strategies for initializing the dictionary estimates \( \tilde{a}_{mn} \). One option is to draw \( \tilde{a}_{mn} \) randomly from the assumed prior on \( a_{mn} \), 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 \( A \), 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 \( A^{(1)} \).

The second alternative is to initialize \( A^{(1)} \) using an appropriately chosen subset of the columns of \( Y \), which is well motivated in the case that there exists a very sparse representation \( X \). For example, if there existed a decomposition \( Y = AX \) in which \( X \) had 1-sparse columns, then the columns of \( A \) would indeed match a subset of the columns of \( Y \) (up to a scale factor). In the general case, however, it is not apriori obvious which columns of \( Y \) to choose, and so we suggest the following greedy heuristic, which aims for a well-conditioned \( A^{(1)} \): select (normalized) columns from \( Y \) 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 \( Y \) are examined before finding \( N \) acceptable candidates, then the process is repeated using a different random order. If repeated re-orderings fail, then \( A^{(1)} \) is initialized using a random draw from \( A \).

C. EM-BiG-AMP

To tune the distributional parameters \( \theta = [\nu^x, \nu^z, \xi]^T \), we can straightforwardly apply the EM approach from Sec. V-A.

For this, we suggest initializing \( \xi = 0.1 \) (since Sec. VIII-E shows that this works well over a wide range of problems) and initializing \( \nu^z_{nl} \) and \( \nu^x \) using a variation on the procedure suggested for MC that accounts for the sparsity of \( x_{nl} \):

\[ \nu^x = \frac{\| P_{\Omega}({Y}) \|_F^2}{\text{SNR}_0 + 1 |\Omega|} \]  

(135)

\[ \nu^z_{nl} = \frac{1}{N\xi} \left[ \| P_{\Omega}({Y}) \|_F^2 |\Omega| - \nu^x \right]. \]  

(136)

D. Avoiding Local Minima

The DL problem is fraught with local minima (see, e.g., 63), 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 the residual error \( \| Z - Y \|_F \) and the average sparsity (as measured by \( \frac{1}{N^2} \sum_{nl} \text{Pr}(x_{nl} \neq 0 | Y = Y) \)) decrease.

\[ \text{As an alternative, if both the previous and current solutions achieve sufficiently small residual error, then only the average sparsity is considered in the comparison.} \]
\textbf{E. Dictionary Learning Experiments}

In this section, we numerically investigate the performance of EM-BiG-AMP for DL, as described in Sec. \textbf{VII}. Comparisons are made with the greedy K-SVD algorithm \cite{63}, the SPAMS implementation of the online approach \cite{65}, and the ER-SpUD(proj) approach for square dictionaries \cite{63}.

1) \textbf{Noiseless Square Dictionary Recovery}: We first investigate recovery of square (i.e., $N = M$) dictionaries from noise-free observations, repeating the experiment from \cite{63}. For this, realizations of the true dictionary $A$ 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 $X$ 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 $Y = AX$, from which the algorithms estimated $A$ and $X$ (up to a permutation and scale). The accuracy of the dictionary estimate $\hat{A}$ was quantified using the relative NMSE metric \cite{63}

$$\text{NMSE}(\hat{A}) \triangleq \min_{J \in J} \frac{\| \hat{A}J - A \|^2_F}{\| A \|^2_F},$$

(137)

where $J$ is a generalized permutation matrix used to resolve the permutation and scale ambiguities.

The subplots on the left of Fig. \textbf{12} show the mean NMSE achieved by K-SVD, SPAMS, ER-SpUD(proj), and EM-BiG-AMP\cite{63} respectively, over 50 problem realizations, for various combinations of dictionary size $N \in \{10, \ldots, 60\}$ and data sparsity $K \in \{1, \ldots, 10\}$, 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 from Sec. \textbf{VII-D} while K-SVD and SPAMS used oracle knowledge to choose the NMSE-minimizing initialization.

The left column in Fig. \textbf{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

The right subplots in Fig. \textbf{12} show the mean value (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.

2) \textbf{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 $Y = Z + W$, where $Z = AX$ and $W$ contained AWGN samples with variance adjusted to achieve an $\text{SNR} = \text{E} \{ \sum_{m,l} |z_{ml}|^2 \} / \text{E} \{ \sum_{m,l} |y_{ml} - z_{ml}|^2 \}$ of 40 dB.

The right subplots in Fig. \textbf{12} show the mean value (over 10 trials) of the relative NMSE from \cite{137} 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

EM-BiG-AMP runs within an order-of-magnitude of the fastest algorithm (SPAMS) and more than two orders-of-magnitude faster than ER-SpUD(proj)\cite{63} for larger dictionaries.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig12.png}
\caption{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.}
\end{figure}
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.

3) 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. 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. 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.

4) Summary: In summary, Figs. 12-14 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 [41] that bears similarity to our scalar-variance BiG-AMP under AWGN-corrupted observations (recall footnote 6). Although their approach gave good results for blind calibration, they report that it was “not able to solve” the DL problem [41]. We attribute EM-BiG-AMP’s success with DL (as evidenced by Figs. 12-14) to the adaptive damping procedure proposed in Sec. IV-B, the initialization procedure proposed in Sec. VIII-B, the EM-learning procedure proposed in Sec. VII-C, and the re-initialization procedure proposed in Sec. VII-D.

IX. Conclusion

In this work, we presented BiG-AMP, an extension of the G-AMP 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 [39] and are very encouraging.

APPENDIX A

Here we derive (31)-(32), which are stated without a detailed derivation in [32]. Recalling (21) and omitting the ml subscripts for brevity, it can be seen that

$$H' (q; \nu^2; y) = \frac{\partial}{\partial q} \log \int p_{y|z}(y|z) \frac{1}{\sqrt{2\pi \nu^2}} \exp \left( -\frac{1}{2\nu^2} (z - \hat{q})^2 \right) dz$$

$$= \frac{\partial}{\partial q} \left( \log \int \exp \left( \log p_{y|z}(y|z) - \frac{z^2}{2\nu^2} \frac{\hat{q}z}{\nu^2} \right) dz - \frac{\hat{q}^2}{2\nu^2} \right)$$

$$= -\frac{\hat{q}}{\nu^2} \log \int \exp (\phi(u) + \hat{q}u) du \quad \text{via} \quad u \triangleright \frac{z}{\nu^2}$$

(138)

for an appropriately defined function $\phi(\cdot)$. Now, defining $p_{u|q}(u|q) \triangleq Z(q)^{-1} \exp(\phi(u) + \hat{q}u)$ with normalization term $Z(q) \triangleq \int \exp (\phi(u) + \hat{q}u) du$, simple calculus yields

$$\frac{\partial}{\partial q} \log Z(q) = E[u|q = \hat{q}]$$

(139)

$$\frac{\partial^2}{\partial q^2} \log Z(q) = \text{var}[u|q = \hat{q}]$$

(140)

Thus, from (138) and (139) it follows that

$$H' (q; \nu^2; y) = -\frac{\hat{q}}{\nu^2} + \int \frac{\exp (\phi(u) + \hat{q}u)}{Z(q)} du$$

$$= -\frac{\hat{q}}{\nu^2} + \frac{z}{\nu^2} \int \frac{\exp (\log p_{y|z}(y|z) - \frac{z^2}{2\nu^2} \frac{\hat{q}z}{\nu^2})}{Z(q)} dz$$

$$= -\frac{\hat{q}}{\nu^2} + \frac{1}{\nu^2} \int \frac{z p_{y|z}(y|z) N(z; \hat{q}, \nu^2)}{Z(q) N(z; \hat{q}, \nu^2)} dz$$

(141)

Equation (31) is then established by applying definitions (32) and (35) to (141).

Similarly, from (138) and (140),

$$-H'' (q; \nu^2; y) = \frac{\partial}{\partial q} \left( \frac{\hat{q}}{\nu^2} - \frac{\partial}{\partial q} \log Z(q) \right) = \frac{1}{\nu^2} - \text{var}[u|q = \hat{q}]$$

$$= \frac{1}{\nu^2} + \frac{1}{\nu^2} \int \left( z - \hat{q} \right)^2 \frac{z p_{y|z}(y|z) N(z; \hat{q}, \nu^2)}{N(z; \hat{q}, \nu^2)} dz.$$

(142)

where $\hat{z}$ is the expectation from (35). Equation (32) is then established by applying the definitions (32) and (35) to (142).

APPENDIX B

Here we explain the approximations (65)-(66). The term neglected in going from (45) to (65) can be written using (31)-(32) as

$$\sum_{m=1}^{M} \nu_{mn}(t) \left( \tilde{z}_{ml}(t) - \nu_{ml}^a(t) \right)^2 + \nu_{ml}^a(t) \left( 1 - \frac{1}{\nu_{ml}^a(t)} \right)$$

$$= \sum_{m=1}^{M} \nu_{mn}(t) \left[ \left( \tilde{z}_{ml}(t) - \frac{1}{\nu_{ml}^a(t)} \right)^2 - 2 \right]$$

(143)

and

$$= \sum_{m=1}^{M} \nu_{mn}(t) \left[ \frac{1}{\nu_{ml}^a(t)} \right]$$

(144)

where the expectations are taken over $z_{ml} \sim p_{z_{ml}|p_{ml}}(\cdot; \nu_{ml}^a(t); \nu^p_{ml}(t))$ from (35). For GAMP, (32) Sec. VLD clarifies that, in the large system limit, under i.i.d priors and scalar variances, the true $z_{ml}$ and the iterates $\tilde{p}_{ml}(t)$ converge empirically to a pair of random variables $(z, \nu)$ that satisfy $p_{z|\nu}(z|\nu) = N(z; \nu^a(t), \nu^p(t))$. This result leads us to believe that the expectation in (144) is approximately univalued when averaged over $m$, and thus (144) is approximately zero-valued. Similar reasoning applies to (66).

ACKNOWLEDGMENT

The authors would like to thank Sundeep Rangan, Florent Krzakala, and Lenka Zdeborová for insightful discussions on various aspects of AMP-based inference. We would also like to thank Subhojit Som, Jeremy Vila, and Justin Ziniel for helpful discussions about EM and turbo methods for AMP.

REFERENCES

[1] 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.
[2] 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).
[3] J. T. Parker and P. Schniter, "Bilinear generalized approximate message passing (BiG-AMP) for matrix completion," Nov. 2012, presented at the Asilomar Conf., (Pacific Grove, CA).
[4] 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
[5] 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/s10107-009-0360-5
[6] Z. Zhou, X. Li, J. Wright, E. Cand`e, and, Y. Ma, "Stable principal component pursuit," in Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on, June 2010, pp. 1518 –1522.
[7] 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/10.1137/09076183X
[8] 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.
[59] R. Salakhutdinov and A. Mnih, “Probabilistic matrix factorization,” in Proc. Neural Inform. Process. Syst. Conf., Vancouver, BC, Dec. 2008, pp. 1257–1264.

[60] 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.

[61] E. J. Candes, 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

[62] R. Rubinstein, A. Bruckstein, and M. Elad, “Dictionaries for sparse representation modeling,” Proceedings of the IEEE, vol. 98, no. 6, pp. 1045–1057, 2010.

[63] 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.

[64] 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.

[65] 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.