Bilinear Generalized Vector Approximate Message Passing

Mohamed Akrout, Anis Housseini, Faouzi Bellili, Member, IEEE, and Amine Mezghani, Member, IEEE

E2-390 E.I.T.C. 75 Chancellor’s Circle Winnipeg, MB, Canada, R3T 5V6.
Emails: {akroutm, houssieia}@myumanitoba.ca, {faouzi.bellili, amine.mezghani}@umanitoba.ca.

Abstract—We introduce the bilinear generalized vector approximate message passing (BiG-VAMP) algorithm which jointly recovers two matrices $U$ and $V$ from their noisy product through a probabilistic observation model. BiG-VAMP provides computationally efficient approximate implementations of both max-sum and sum-product loopy belief propagation (BP). We show how the proposed BiG-VAMP algorithm recovers different types of structured matrices and overcomes the fundamental limitations of other state-of-the-art approaches to the bilinear recovery problem, such as BiG-AMP, BAD-VAMP and LowRAMP. In essence, BiG-VAMP applies to a broader class of practical applications which involve a general form of structured matrices. For the sake of theoretical performance prediction, we also conduct a state evolution (SE) analysis of the proposed algorithm and show its consistency with the asymptotic empirical mean-squared error (MSE). Numerical results on various applications such as matrix factorization, dictionary learning, and matrix completion demonstrate unambiguously the effectiveness of the proposed BiG-VAMP algorithm and its superiority over state-of-the-art algorithms. Using the developed SE framework, we also examine (as one example) the phase transition diagrams of the matrix completion problem, thereby unveiling a low detectability region corresponding to the low signal-to-noise ratio (SNR) regime.

Index Terms—Bayesian inference, approximate message passing, bilinear structured matrix recovery, inference algorithms, matrix factorization, dictionary learning, matrix completion.

I. INTRODUCTION

A. Background and related work

We consider an observation matrix, $Y \in \mathbb{R}^{N \times M}$, obtained from the following generalized bilinear model:

$$p_{Y|Z}(Y|Z) = \prod_{i=1}^{N} \prod_{j=1}^{M} p_{y_{ij}|z_{ij}}(y_{ij}|z_{ij}) \quad \text{with} \quad Z = UV^T,$$

(1)

where $U$ and $V$ are two unknown matrices in $\mathbb{R}^{N \times r}$ and $\mathbb{R}^{M \times r}$, respectively. The goal is to recover $U$ and $V$ based on the knowledge of $Y$ and the model in (1). The latter applies to a myriad of problems ranging from noisy dictionary learning [1], matrix completion [2], and sparse PCA [3], to matrix factorization [4], low-rank matrix reconstruction [5], and subgraph estimation [6], just to name a few. A special relevant case of the generalized bilinear model in (1) is:

$$Y = \phi(UV^T + W),$$

(2)

in which $W \in \mathbb{R}^{N \times M}$ is an additive white Gaussian noise matrix whose entries are assumed to be mutually independent with mean zero and variance $\gamma_w^{-1}$, i.e., $w_{ij} \sim \mathcal{N}(0, \gamma_w^{-1})$. While convex relaxation of the bilinear recovery problem under the observation model in (2) is possible in some cases using the augmented Lagrange multiplier method (ALMM) [7], different non-convex formulations have been investigated in some special cases over the last decade based on:

1) the alternating direction method of multipliers (ADMM) [8],
2) the variational sparse Bayesian learning (VSBL) method [9],
3) the approximate message passing (AMP) paradigm which is discussed hereafter in some depth in order to put our contribution in a proper perspective.

In fact, AMP-based computational information/data processing have attracted a lot of interest in different fields since the early introduction of the AMP algorithm in [10] within the compressed sensing (CS) framework. More specifically, in a typical CS problem, one is interested in recovering an unknown sparse vector, $x \in \mathbb{R}^{N}$, from its noisy linear measurements/observations:

$$y = Ax + w,$$

(3)

wherein $A \in \mathbb{R}^{M \times N}$ (with $M \ll N$) is a known sensing matrix. AMP strikes a proper balance between reconstruction performance and computational complexity as compared to traditional convex optimization-based and iterative soft thresholding algorithms [11]. The AMP algorithm was later extended in [12] to generalized linear models of the form:

$$p_{y|x}(y|x) = \prod_{m=1}^{M} p_{y_m|x_m}(y_m|x_m) \quad \text{with} \quad z = Ax.$$

(4)

Aside from handling nonlinear transformations, the advantage of the generalized AMP (GAMP) algorithm over its AMP predecessor lies in its ability to accommodate statistical priors on the sparse vector $x$. From another perspective, the performance of both AMP and GAMP can be rigorously tracked by a set of scalar update equations, known as state evolution (SE) [13], [14].
in statistics or the cavity method in statistical physics [15]. One limitation of AMP and GAMP, however, is that they often diverge if the sensing matrix, \( A \), is ill-conditioned and/or has a non-zero mean. It is precisely in this context that the vector AMP (VAMP) algorithm has been recently introduced and rigorously analyzed in [16]–[18]. Although, there is no theoretical guarantees that VAMP will always converge, there is strong empirical evidence that it is more resilient to mean-perturbed or badly conditioned sensing matrices \( A \), provided that the latter is right-orthogonally invariant [18].

To date, extensions of the original AMP algorithm to the bilinear recovery problem were already made in [5] and [20] within the context of low-rank matrix reconstruction. In this context, the so-called LowRAMP algorithm introduced in [20] extends the method in [5] to non-Gaussian likelihoods and establishes the associated SE analysis. It does so by simplifying the BP messages in [5] via second-order Taylor series approximations which become more accurate in the limit of large \( N \). However, the involved approximations hold in the per-measurement low-SNR regime only [i.e., \( \text{SNR} = O(\frac{1}{N}) \)] and, hence, LowRAMP is not suitable for other types of measurement processes such as quantization and matrix sub-sampling. Moreover, the “low-rank” assumption which is critical in both [5] and [20] precludes a large number of practical applications which involve a general-rank decomposition of structured matrices rather than a low-rank decomposition of unstructured matrices.

GAMP itself was also extended to the bilinear case in [21] thereby leading to the so-called bilinear generalized AMI (BiG-AMP) algorithm. Although being completely oblivious to the low-rank assumption, BiG-AMP inherits all the GAMP-related convergence issues and was developed for separable priors on both \( U \) and \( V \) matrices. To accommodate a larger class of \( V \) matrices in [1]. Sarkar et al. made an attempt in [23] to generalize the VAMP framework to bilinear recovery problems and the algorithm introduced therein was called Bilinear Adaptive VAMP (BAd-VAMP). In essence, BAd-VAMP is a ping-pong-like approach which reconstructs both \( U \) and \( V \) matrices by alternating between i) the expectation-maximization (EM) algorithm [24] to find the maximum-likelihood (ML) estimate of \( U \) and ii) the VAMP algorithm [18] to find the minimum mean-squared error (MMSE) estimate of \( V \). Unfortunately, due to the use of the EM algorithm, BAd-VAMP does not accommodate general priors on the matrix \( U \). For instance, trying to enforce a binary or sparsity prior on \( U \) renders the E-step of the EM algorithm computationally prohibitive. To overcome all the aforementioned limitations, this paper introduces a new algorithm along with its state evolution analysis to solve a broader class of the bilinear recovery problems as modelled in [1].

B. Contributions

This paper builds upon the prior work in [5] and provides a broader solution to the bilinear recovery problem under different structured matrices beyond the “low-rank” assumption.

1It is worth mentioning here that VAMP was independently derived by two research groups in [16] and [19] but under two different names, i.e., VAMP and orthogonal AMP (OAMP), respectively.

2The reader is also referred to the parametric version of BiG-AMP in [22].

Our approach for bilinear recovery does not alternate between the EM and VAMP algorithms, but rather relies entirely on message passing and is dubbed bilinear generalized VAMP (BiG-VAMP). The BiG-VAMP approach that we propose in this work enables the use of arbitrary priors on both \( U \) and \( V \), thereby allowing the exploitation of other matrix structures such as finite-alphabet, binarity, sparsity, constant-modulus, assignment, etc. The proposed BiG-VAMP algorithm is suitable to a broader class of bilinear recovery problems that i) cover more general prior distributions on the unknown matrices (unlike BiG-AMP) and ii) does not rely on the use of the EM algorithm with automated hyperparameter tuning to estimate \( U \) (unlike BAd-VAMP). The key differences between BiG-VAMP and LowRAMP are, however, as follows:

1) BiG-VAMP provides a systematic way for handling non-linear outputs without the “low-SNR” assumption which does not hold, e.g., in the matrix completion problem. Moreover, BiG-VAMP allows maximum posteriori bilinear reconstruction under non-differentiable output functions such as quantization, perceptron activation, selection, and phase-retrieval, etc.

2) BiG-VAMP applies to a broader class of practical applications which involve a general-rank decomposition of structured matrices in addition to a low-rank decomposition of unstructured matrices.

Much like BiG-AMP, BiG-VAMP is also applicable to maximum posteriori (MAP) and MMSE inference problems alike, as will be explained later on. Furthermore, it comes with theoretical performance guarantees, established in Section IV, that validate its superiority against state-of-the-art BiG-AMP, BAd-VAMP, and LowRAMP algorithms.

Notation

We use Sans Serif font (e.g., \( x \)) for random variables and Serif font (e.g., \( x \)) for its realizations. We use boldface lowercase letters for vectors (e.g., \( \mathbf{x} \) and \( \mathbf{x} \)) and boldface uppercase letters for matrices (e.g., \( \mathbf{X} \) and \( \mathbf{X} \)). Vectors are in column-wise orientation by default. Given any matrix \( \mathbf{X} \), we use \( x_i \) and \( x_{ij} \) to denote its \( i \)th column and \( i-j \)th entry, respectively. We also denote the \( i \)th component of a vector \( x \) as \( [x]_i \) or \( x_i \). The operator \( \text{diag}(X) \) stacks the diagonal elements of \( X \) in a vector while \( I \) stands for the identity matrix. The operator \( \text{tr}(X) \) returns the sum of the diagonal elements of \( \mathbf{X} \). We also use \( p_\theta(x; \theta), p_\theta(x; \bar{\theta}), \) and \( p_{\mathbf{X}; \hat{\theta}} \) to denote the pdf of random variables/vectors/matrices; as being parameterized by a set of parameters \( \theta \). Moreover, \( \mathcal{N}(x; \mathbf{x}, \mathbf{R}) \) stands for the multivariate Gaussian pdf of any random vector \( x \) with mean \( \mathbf{x} \) and covariance matrix \( \mathbf{R} \). We use \( \sim \) and \( \propto \) as short-hand notations for “distributed according to” and “proportional to”, respectively. We also use \( E[x|d(x)] \) to denote the expectation of \( x \sim d(x) \) and \( d(x) \) refers to the Dirac delta distribution. Moreover, \( \langle x \rangle \) and \( \langle X \rangle \) return the (empirical) average values of vectors and matrices, i.e., \( \langle x \rangle \triangleq \frac{1}{N} \sum_{i=1}^{N} x_i \) for \( x \in \mathbb{R}^N \) and \( \langle X \rangle \triangleq \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} x_{ij} \) for \( X \in \mathbb{R}^{N \times M} \).

Finally, the symbol \( \odot \) denotes the Hadamard (i.e., elementwise) product between any two matrices.
II. BACKGROUND ON THE LOW-RANK MATRIX RECONSTRUCTION

In this section, we briefly review the main results of the prior work on low-rank matrix reconstruction in [5] at the detail needed for a comprehensive exposition of BiG-VAMP. We emphasize, however, the fact that borrowing such results does not restrict the proposed BiG-VAMP algorithm to the bilinear low-rank matrix recovery as is the case in [5]. Consider the bilinear recovery of two random independent matrices $U = [u_1, \ldots, u_N]^T \in \mathbb{R}^{N \times r}$ and $V = [v_1, \ldots, v_M]^T \in \mathbb{R}^{M \times r}$ from a linear noisy observation $Y = UV^T + W \in \mathbb{R}^{N \times M}$. Given some common priors, $p_u(u_i)$ and $p_v(v_j)$, on the vectors $u_i$ and $v_j$, respectively, the goal of BP is to approximate their joint posterior distribution:

$$p_{u|Y}(u_i, v_j|Y; \gamma_w^{-1}, \beta) \propto \prod_{i=1}^N \prod_{j=1}^M p_{y_{ij}|u,v}(y_{ij}|u_i, v_j; \gamma_w^{-1})^\beta p_u(u_i)^\beta p_v(v_j)^\beta,$$  

wherein $\beta$ is a parameter introduced here to treat the MMSE ($\beta = 1$) and MAP ($\beta = +\infty$) inference problems in a unified framework. We also assume the priors, $p_u(U)$ and $p_v(V)$, on the unknown matrices of $U$ and $V$ to be row-wise separable, i.e., $p_u(U) = \prod_{i=1}^N p_u(u_i)$ and $p_v(V) = \prod_{j=1}^M p_v(v_j)$. Fig. 1 depicts the factor graph associated to (5) with variable nodes, $u_i$ and $v_j$, their prior factor nodes, $p_u(u_i)^\beta$ and $p_v(v_j)^\beta$, and the labels, $f_{ij}$, which we use as a shorthand notations for the factor nodes:

$$f(u_i, v_j) \triangleq p_{y_{ij}|u,v}(y_{ij}|u_i, v_j; \gamma_w^{-1})^\beta,$$  

(6)

By taking $\beta = 1$ (i.e., minimum mean square error estimation), $p_{u|Y}(u_i, v_j|y_{ij}; \beta)$ reduces to the true joint posterior $p_{u,v|y_{ij}}(u_i, v_j|y_{ij})$. In the limit $\beta \to \infty$ (i.e., maximum a posteriori estimation), however, it concentrates on the maxima of $p_{u,v|y_{ij}}(u_i, v_j|y_{ij})$. Using the message derivation rules of loopy BP, the four messages defined in Fig. 1, are expressed as follows (where $t$ stands for the iteration index):

$$\mu_{i\to j},t (u_i) \propto \int f(u_i, v_j) \nu_{j\to i},t-1 (v_j) \, dv_j,$$  

(8)

$$\mu_{i\to j},t+1 (u_i) \propto p_u(u_i)^\beta \prod_{l \neq j} \mu_{l\to i},t (u_i),$$  

(9)

$$\nu_{i\to j},t (v_j) \propto \int f(u_i, v_j) \mu_{i\to j},t-1 (u_i) \, du_i,$$  

(10)

$$\nu_{i\to j},t+1 (v_j) \propto p_u(v_j)^\beta \prod_{k \neq i} \nu_{j\to k},t (v_j).$$  

(11)

Assuming $v_j \sim \nu_{i\to j},t (v_j)$ with mean $\mu_{i\to j},t (u_i)$ and covariance matrix $\beta^{-1} R_{i\to j},t$ it was shown in [5] by virtue of the central limit theorem (CLT) that the product of incoming messages, $\prod_{t \neq t} \mu_{i\to j},t (u_i)$, to any variable node $u_i$ from all factor nodes $\{f_{il}\}_{l \neq j}$ can be approximated by a Gaussian random variable with mean $b_{u,i\to(i,j),t}$ and precision $\beta A_{u,i\to(i,j),t}$:

$$b_{u,i\to(i,j),t} = \gamma_w \sum_{l \neq j} v_{i,l}^T \hat{v}_{l\to(i,j),t},$$  

(12a)

$$A_{u,i\to(i,j),t} = \gamma_w \sum_{l \neq j} (\hat{v}_{l\to(i,j),t} - \beta^{-1} R_{i\to j},t)^T R_{i\to j},t. $$  

(12b)

In other words, by dropping the normalization factor that does not depend on $u_i$, we have:

$$\prod_{t \neq j} \mu_{i\to j},t (u_i) \propto \exp \left(-\frac{\beta}{2} \hat{v}_{j\to i}^T \Lambda_{u,i\to(i,j),t} u_i + \beta u_i^T b_{u,i\to(i,j),t}\right).$$  

(13)

The inherent symmetry among the variable nodes $u_i$ and $v_j$ yields an equivalent Gaussian approximation for $\prod_{k \neq i} \nu_{(k\to j),t} (v_j)$ under the density of $u_k \sim \mu_{k\to(i,j),t} (u_k)$ with mean $\mu_{k\to(i,j),t}$ and covariance matrix $\beta^{-1} R_{u,k\to(i,j),t}$. That is to say:

$$\prod_{k \neq i} \nu_{(k\to j),t} (v_j) \propto \exp \left(-\frac{\beta}{2} v_j^T \Lambda_{v,j\to(i,j),t} v_j + \beta v_j^T b_{v,j\to(i,j),t}\right).$$  

(14)
with

\[ b_{u,j}\to(i,j),t} = \gamma_u \sum_{k \neq i} y_{k,j} \hat{u}_{k\to(k,j),t}, \]  
\[ A_{u,j}\to(i,j),t} = \gamma_u \sum_{k \neq i} \left( \hat{u}_{k\to(k,j),t} \hat{u}_{k\to(k,j),t}^T + \beta^{-1} R_{u,k\to(k,j),t} \right) \]  
\[ - \gamma_w y_{k,j}^2 R_{u,k\to(k,j),t}, \]  
\[ \Lambda_{u,i,t} = \gamma_u \sum_{l} \left( \hat{v}_{i\to(i,l),t} \hat{v}_{i\to(i,l),t}^T + \left( \beta^{-1} - \gamma_w y_{i,l}^2 \right) R_{v,i,l,t} \right). \]  
\[ \Lambda_{v,i,t} = \gamma_w \sum_{k} \left( \hat{v}_{k\to(k,i),t} \hat{v}_{k\to(k,i),t}^T + \left( \beta^{-1} - \gamma_w y_{k,i}^2 \right) R_{v,k,i,t} \right). \]

Pictorially, the messages given in (13) and (14) are shown in Fig. 2 as messages 1 and 2 sent by all factor nodes \( \{f_{j'}\}_{j'=1,j\neq j'}^N \) and \( \{f_{ij'}\}_{j'=1,j\neq j'}^N \) to \( u_i \) and \( v_j \), respectively. Moreover, the mean

\[ \mathcal{N}(u_i; \hat{u}_{i\to(i,j),t}, \Lambda_{u,i-t\to(i,j),t}), \]  
\[ \mathcal{N}(v_j; \hat{v}_{j\to(j,i),t}, \Lambda_{v,j-t\to(j,i),t}) \]

values, \( \hat{u}_{i\to(i,j),t+1} \) and \( \hat{v}_{j\to(j,i),t+1} \), as well as the covariance matrices, \( \beta^{-1} R_{u,i\to(i,j),t+1} \) and \( \beta^{-1} R_{v,j\to(j,i),t+1} \), of messages 3 and 4, respectively, are given by:

\[ \hat{u}_{i\to(i,j),t+1} = f_u(b_{u,i\to(i,j),t}, \Lambda_{u,i-t\to(i,j),t}), \]  
\[ \hat{v}_{j\to(j,i),t+1} = f_v(b_{v,j\to(j,i),t}, \Lambda_{v,j-t\to(j,i),t}), \]  
\[ R_{u,i\to(i,j),t+1} = \nabla b_{u,i\to(i,j),t} f_u(b_{u,i\to(i,j),t}, \Lambda_{u,i-t\to(i,j),t})^T, \]  
\[ R_{v,j\to(j,i),t+1} = \nabla b_{v,j\to(j,i),t} f_v(b_{v,j\to(j,i),t}, \Lambda_{v,j-t\to(j,i),t})^T. \]

where

\[ f_u(b, \Lambda^{-1}) = \int p_u(u) u^{\beta} \mathcal{N}(u; \Lambda^{-1} b, \beta^{-1} \Lambda^{-1}) du, \]  
\[ f_v(v, \Lambda^{-1}) = \int p_v(v) v^{\beta} \mathcal{N}(v; \Lambda^{-1} b, \beta^{-1} \Lambda^{-1}) dv, \]  
and the nabla operator, \( \nabla x \), with respect to any \( n \)-dimensional vector, \( x = [x_1, x_2, \ldots, x_n]^T \), is given by:

\[ \nabla x = \left[ \frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \ldots, \frac{\partial}{\partial x_n} \right]^T. \]

To reduce the complexity in the number of computed messages, the \( (i,j) \)-dependent quantities in (12) and (15), are replaced by the following \( (i,j) \)-independent (i.e., broadcast) ones:

\[ b_{u,i,t} = \gamma_u \sum_{l} y_{l,i} \hat{u}_{l\to(i,l),t}, \]  
\[ A_{u,i,t} = \gamma_u \sum_{l} \left( \hat{u}_{l\to(i,l),t} \hat{u}_{l\to(i,l),t}^T + \left( \beta^{-1} - \gamma_w y_{l,i}^2 \right) R_{u,l,i,t} \right). \]  
\[ b_{v,j,t} = \gamma_w \sum_{k} y_{j,k} \hat{v}_{k\to(j,k),t}, \]  
\[ A_{v,j,t} = \gamma_w \sum_{k} \left( \hat{v}_{k\to(j,k),t} \hat{v}_{k\to(j,k),t}^T + \left( \beta^{-1} - \gamma_w y_{k,j}^2 \right) R_{v,k,j,t} \right). \]

after approximating the covariance matrices, \( \beta^{-1} R_{u,i\to(i,j),t} \) and \( \beta^{-1} R_{v,j\to(j,i),t} \), involved in (12) and (15), by broadcast covariances, \( \beta^{-1} R_{u,i,t} \) and \( \beta^{-1} R_{v,j,t} \), respectively, with a vanishing error order \( O(M^{-1}) \). By recalling (16)–(19), the underlying broadcast means and the associated broadcast covariances are thus given by:

\[ \hat{u}_{i\to(i,j),t} = f_u(b_{u,i\to(i,j),t}, \Lambda_{u,i-t\to(i,j),t}), \]  
\[ R_{u,i,t} = \nabla b_{u,i\to(i,j),t} f_u(b_{u,i\to(i,j),t}, \Lambda_{u,i-t\to(i,j),t})^T, \]  
\[ \hat{v}_{j\to(j,i),t} = f_v(b_{v,j\to(j,i),t}, \Lambda_{v,j-t\to(j,i),t}), \]  
\[ R_{v,j,t} = \nabla b_{v,j\to(j,i),t} f_v(b_{v,j\to(j,i),t}, \Lambda_{v,j-t\to(j,i),t})^T. \]

Moreover, the \( (i,j) \) posterior means, \( \hat{u}_{i\to(i,j),t} \) and \( \hat{v}_{j\to(j,i),t} \), are related to their broadcast versions, \( \hat{u}_{i,t} \) and \( \hat{v}_{j,t} \), through small Osanger correction terms of order \( O(M^{-1/2}) \) as follows:

\[ \hat{u}_{i\to(i,j),t} \approx \hat{u}_{i,t} - \gamma_w y_{i,j} R_{u,i,t} \hat{v}_{j,t-1}, \]  
[Osanger correction term on \( u_i \)]
\[ \hat{v}_{j\to(j,i),t} \approx \hat{v}_{j,t} - \gamma_w y_{j,i} R_{v,j,t} \hat{u}_{i,t-1}, \]  
[Osanger correction term on \( v_j \)]

Yet, the correction terms in (29) are taken into account during the calculation of \( b_{u,i,t} \) and \( b_{v,j,t} \) only. For the computation of \( \Lambda_{u,i,t} \) and \( \Lambda_{v,j,t} \), however, one replaces \( \hat{u}_{i\to(i,j),t} \) by \( \hat{u}_{i,t} \) and \( \hat{v}_{j\to(j,i),t} \) by \( \hat{v}_{j,t} \) after ignoring terms of vanishing order as \( M \) and \( N \) grow large. The algorithm in [5] also relies on the following low-SNR approximation

\[ y_{k,j}^2 \approx \mathbb{E}[y_{k,j}^2] \approx \gamma_w^{-1}. \]  

which is used in both (23b) and (24b). We emphasize, however, the fact that the low-SNR regime is mainly conceivable in presence of very-low-rank structures with a fully observed matrix \( Y = UV^T + W \).

In conclusion, the algorithmic steps of the bilinear recovery technique introduced in [5] are summarized in Algorithm 1. We refer the reader to the supplementary materials of [5] for further details. The limitation of this method, however, lies in the intractable multi-dimensional integrals in (20) and (21) which can be evaluated only in some specific priors, \( p_u(\cdot) \) and \( p_v(\cdot) \),
the Gaussian approximation of the extrinsic information as a key means to handle general priors on both $U$ and $V$ matrices in bilinear models. Finally, we extend the results to the generalized bilinear models.

### A. Bilinear Vector Approximate Message Passing (Bi-VAMP) with general rank

In this case, the data matrix, $Y$, is obtained from the bilinear observation model in (2) with $\phi_i$ being the identity, i.e., $\phi(x) = x$, $\forall x \in \mathbb{R}$. That is to say:

$$ Y = UV^T + W, $$

(31)

where the noise components, $w_{ij}$, are mutually independent and Gaussian distributed with zero mean and variance $\gamma_w^{-1}$. All the algorithmic steps of Bi-VAMP which will be explained in this section are summarized in Algorithm 2 after excluding the update equations pertaining to the “generalized output step” (i.e., lines 26-32), while replacing $\gamma_{Z}^{-1}$ by $\gamma_w^{-1}$ and $Z^*$ by $Y$.

To sidestep the problem of computing the intractable integrals in (20) and (21) during the evaluation of the mean and variance of messages (3) and (4) in Fig. 2, we proceed as follows. We rewrite the original posterior factorization in (5) by splitting $u_i$ (resp. $v_j$) into two identical variables with equality constraints in between, i.e., $u_i^\perp = u_i^-$ (resp. $v_j^\perp = v_j^+$), thereby yielding the following equivalent factorization:

$$ p_{u_i^\perp, u_i^+, v_j^\perp, v_j^+} \propto \prod_{i=1}^{N} \prod_{j=1}^{M} p_{y_{ij}|u_i^-, v_j^+, \gamma_w^{-1}}^{\beta} \prod_{i=1}^{N} p_{u_i^+}^{\beta} \prod_{j=1}^{M} p_{v_j^+}^{\beta}, $$

(32)

The new factorization in (32) transforms the original factor graph in Fig. 2 into the new one depicted in Fig. 3. To handle the newly introduced equality constraints, the new variables $u_i^\perp$, $u_i^+$, $v_j^\perp$, and $v_j^+$ are rather regarded as processing nodes which exchange scalar messages/beliefs in the form of component-wise (i.e., decoupled) Gaussian densities. The beliefs provided by $u_i^\perp$ and $v_j^\perp$ on $u_i^+$ and $v_j^+$ (and vice versa) are known as the extrinsic information and are modelled by the Gaussian messages (1), (2), (3) and (4) in Fig. 3 whose parameters will be calculated later in this section.

The decoupling in the messages on each side of the equality nodes results in two simple types of denoising functions, namely $f_{\ell_u}(\cdot)$ [resp. $f_{\ell_v}(\cdot)$] and $g_{\ell_u}(\cdot)$ [resp. $g_{\ell_v}(\cdot)$] to recover $u_i^\perp$ (resp. $v_j^\perp$) and $u_i^+$ (resp. $v_j^+$). Such decoupled message passing is made possible by ignoring the off-diagonal elements of the error covariance matrices calculated on the side of $u_i^\perp$ and $v_j^\perp$ nodes. Although the integrals of the denoising functions, $f_{\ell_u}(\cdot)$ and $f_{\ell_v}(\cdot)$, in (20) and (21) are still required for Bi-VAMP, they are now evaluated in closed form after replacing the actual priors, $p_{\ell_u}(\cdot)$ and $p_{\ell_v}(\cdot)$, with the extrinsic Gaussian messages (3) and (4), respectively. Messages (1) and (2) are updated in lines 24-27 of Algorithm 2 in a matrix form (i.e., $B_{U,t}$, $B_{V,t}$, $\Lambda_{U,t}$, and $\Lambda_{V,t}$)
Algorithm 2 Bi-G-VAMP

Require : Matrix $Y \in \mathbb{R}^{N \times M}$, temperature parameter $\beta$; precision tolerance ($\epsilon = 10^{-6}$); maximum number of iterations ($T_{\text{max}}$); two denoisers $g_\gamma(\cdot)$ and $g_{\hat{\gamma}}(\cdot)$ from (17) and (18); (noise precision $\gamma_0$ only for Bi-VAMP, i.e., under bilinear observation models)

1: Initialize

2: $t \leftarrow 1$

3: repeat

   I. Approximate Bi-LMMSE step

   4: $B_{U,t} = \gamma_{2,t}^{-1} \left( Z_{e,t} V_{e,t} - M \gamma_{2,t}^t U_{p,t-1} V_{p,t-1} \right) R_{V_{p,t} V_{p,t}}^{-1} \left( \tilde{Z}_{e,t} \otimes Z_{e,t} \right)$

   5: $A_{U,t} = -\gamma_{1,t}^{-1} \left( V_{p,t} T_{e}^t V_{p,t} + \frac{1}{2} R_{V_{p,t} V_{p,t}}^{-1} M \gamma_{2,t}^t Z_{e,t} \otimes Z_{e,t} \right)$

   6: $B_{V,t} = \gamma_{2,t}^{-1} \left( Z_{e,t} V_{e,t} - N \gamma_{2,t}^t U_{p,t-1} V_{p,t-1} \right) R_{V_{p,t} V_{p,t}}^{-1} \left( \tilde{Z}_{e,t} \otimes Z_{e,t} \right)$

   7: $A_{V,t} = \gamma_{2,t}^{-1} \left( U_{p,t} V_{p,t} + N \gamma_{2,t} U_{p,t-1} V_{p,t-1} \right) R_{V_{p,t} V_{p,t}}^{-1} \left( \tilde{Z}_{e,t} \otimes Z_{e,t} \right)$

   II. Denoising step

   8: $R_{U_{p,t} U_{p,t}}^{-1} = \left( \gamma_{1,t}^{-1} I + A_{U,t} \right)^{-1}$

   9: $\hat{U}_{p,t+1} = \left( B_{U,t} + \gamma_{1,t} R_{U_{p,t} U_{p,t}}^{-1} \right) R_{U_{p,t} U_{p,t}}^{-1}$

   10: $R_{V_{p,t} V_{p,t}}^{-1} = \left( \gamma_{2,t}^{-1} I + A_{V,t} \right)^{-1}$

   11: $\hat{V}_{p,t+1} = \left( B_{V,t} + \gamma_{2,t} R_{V_{p,t} V_{p,t}}^{-1} \right) R_{V_{p,t} V_{p,t}}^{-1}$

   III. Generalized output

   12: $\gamma_{U_{p,t} U_{p,t}}^{-1} \left( B_{U,t} + \gamma_{1,t} R_{U_{p,t} U_{p,t}}^{-1} \right) R_{U_{p,t} U_{p,t}}^{-1}$

   13: $\gamma_{V_{p,t} V_{p,t}}^{-1} \left( B_{V,t} + \gamma_{2,t} R_{V_{p,t} V_{p,t}}^{-1} \right) R_{V_{p,t} V_{p,t}}^{-1}$

   14: $\gamma_{U_{p,t} U_{p,t}}^{-1} \left( B_{U,t} + \gamma_{1,t} R_{U_{p,t} U_{p,t}}^{-1} \right) R_{U_{p,t} U_{p,t}}^{-1}$

   15: $\gamma_{V_{p,t} V_{p,t}}^{-1} \left( B_{V,t} + \gamma_{2,t} R_{V_{p,t} V_{p,t}}^{-1} \right) R_{V_{p,t} V_{p,t}}^{-1}$

   16: $\gamma_{U_{p,t} U_{p,t}}^{-1} \left( B_{U,t} + \gamma_{1,t} R_{U_{p,t} U_{p,t}}^{-1} \right) R_{U_{p,t} U_{p,t}}^{-1}$

   17: $\tilde{V}_{e,t+1} = \gamma_{V_{p,t} V_{p,t}}^{-1} \left( \tilde{V}_{p,t+1} \tilde{V}_{e,t+1} - \gamma_{V_{p,t} V_{p,t}} \tilde{V}_{e,t+1} \right)$

   18: $\forall i :$ update the $i$th row of $\tilde{U}_{p,t} + \gamma_{U_{p,t} U_{p,t}}^{-1}$ to $\gamma_{U_{p,t} U_{p,t}}^{-1} \left( \hat{U}_{p,t+1} \gamma_{U_{p,t} U_{p,t}}^{-1} \right)$

   19: $\forall j :$ update the $j$th column of $\tilde{V}_{p,t+1} \gamma_{V_{p,t} V_{p,t}}^{-1}$ to $\gamma_{V_{p,t} V_{p,t}}^{-1} \left( \hat{V}_{p,t+1} \gamma_{V_{p,t} V_{p,t}}^{-1} \right)$

   20: $\gamma_{U_{p,t} U_{p,t}}^{-1} \gamma_{V_{p,t} V_{p,t}}^{-1} \tilde{V}_{e,t+1} \gamma_{V_{p,t} V_{p,t}} \tilde{V}_{e,t+1} \gamma_{V_{p,t} V_{p,t}}^{-1}$

   21: $\gamma_{U_{p,t} U_{p,t}}^{-1} \gamma_{V_{p,t} V_{p,t}}^{-1} \tilde{V}_{e,t+1} \gamma_{V_{p,t} V_{p,t}} \tilde{V}_{e,t+1} \gamma_{V_{p,t} V_{p,t}}^{-1}$

   22: $\gamma_{U_{p,t} U_{p,t}}^{-1} \gamma_{V_{p,t} V_{p,t}}^{-1}$

   23: $\gamma_{V_{p,t} V_{p,t}}^{-1}$

   24: $\gamma_{U_{p,t} U_{p,t}}^{-1} \gamma_{V_{p,t} V_{p,t}}^{-1}$

   25: $\gamma_{U_{p,t} U_{p,t}}^{-1} \gamma_{V_{p,t} V_{p,t}}^{-1}$

   26: $\gamma_{U_{p,t} U_{p,t}}^{-1} \gamma_{V_{p,t} V_{p,t}}^{-1}$

   27: $\gamma_{U_{p,t} U_{p,t}}^{-1} \gamma_{V_{p,t} V_{p,t}}^{-1}$

   28: $\gamma_{U_{p,t} U_{p,t}}^{-1} \gamma_{V_{p,t} V_{p,t}}^{-1}$

   29: $\gamma_{U_{p,t} U_{p,t}}^{-1} \gamma_{V_{p,t} V_{p,t}}^{-1}$

   30: $\gamma_{U_{p,t} U_{p,t}}^{-1} \gamma_{V_{p,t} V_{p,t}}^{-1}$

   31: $\gamma_{U_{p,t} U_{p,t}}^{-1} \gamma_{V_{p,t} V_{p,t}}^{-1}$

   32: $\gamma_{U_{p,t} U_{p,t}}^{-1} \gamma_{V_{p,t} V_{p,t}}^{-1}$

   33: $\gamma_{U_{p,t} U_{p,t}}^{-1} \gamma_{V_{p,t} V_{p,t}}^{-1}$

   34: until $(\|\hat{U}_{p,t+1} - \tilde{U}_{p,t} \|^2 + \|\hat{V}_{p,t+1} - \tilde{V}_{p,t} \|^2) \leq \epsilon (\|\hat{U}_{p,t} \|^2 + \|\hat{V}_{p,t} \|^2)$ or $(t > T_{\text{max}})$

35: return $\hat{U}_{p,t+1}, \hat{V}_{p,t+1}$

as discussed in Section II.

Note, however, that Bi-VAMP avoids the approximation in (30) which is valid for the very-low-rank structure only. To that end, Bi-VAMP incorporates explicitly the contribution of all $\gamma_{e,j}^t$ in (23) and (24b) during the computation of the broadcast precision matrices $A_{U,t}$ and $A_{V,t}$, respectively.

The posterior estimates of $U_t$ and $V_t$ and their error covariance matrices are updated in closed forms as shown in lines 8-11 of Algorithm 2. The resulting regularized matrix inverse structure therein suggests that the updates in lines 8-11 are in essence performing approximate bi-LMMSE recovery of $U$ and $V$ under Gaussian prior information. Based on the i.i.d. assumption, we further reduce the posterior covariance matrices to common scalar variances obtained by simply averaging their diagonal entries (see lines 12 and 15) in Algorithm 2 while ignoring the off-diagonal part. As a result, the posterior covariance matrices are given by the scaled identities, $\gamma_{U_{p,t} U_{p,t}}^{-1} I$ and $\gamma_{V_{p,t} V_{p,t}}^{-1} I$. Such approximation of messages by their means and scalar variances is a common practice in the message passing paradigm, also known as expectation propagation (EP) principle.[6] After computing the posterior estimates and the associated common precision (i.e., $\gamma_{U_{p,t} U_{p,t}}^{-1}$ and $\gamma_{V_{p,t} V_{p,t}}^{-1}$) of the equality node. Formally speaking, this amounts to approximating the posterior messages by Gaussian distributions with the same means and variances (i.e., $\mathcal{N}(u_t; \mu_t, \beta^{-1} \gamma_{U_{p,t} U_{p,t}}^{-1} I)$ and $\mathcal{N}(v_t; \mu_t, \beta^{-1} \gamma_{V_{p,t} V_{p,t}}^{-1} I)$) and extracting the extrinsic messages $\gamma_{e,j}^t$ (resp. $\gamma_{e,j}^t$) before returning the extrinsic messages (11) (resp. (12)) to the other side of the equality node. In error control codes literature, the concept of EP is also known as the turbo principle.

[6] In error control codes literature, the concept of EP is also known as the turbo principle.
These extrinsic precisions and means are updated in lines 13, 14, related as follows:

\[
\gamma_{U_{i,t+1}} = \gamma_{U_{i,t}} - \gamma_{U_i,t}, \\
\gamma_{V_{i,t+1}} = \gamma_{V_{i,t}} - \gamma_{V_i,t}, \\
\widehat{u}_{i,t+1} = \gamma_{U_{i,t+1}}^{-1}/\gamma_{U_{i,t}}^{-1} + \gamma_{U_{i,t}}^{-1} = \gamma_{U_{i,t+1}}^{-1}/\gamma_{U_{i,t}}^{-1} + \gamma_{U_{i,t}}^{-1}, \\
\widehat{v}_{i,t+1} = \gamma_{V_{i,t+1}}^{-1}/\gamma_{V_{i,t}}^{-1} + \gamma_{V_{i,t}}^{-1} = \gamma_{V_{i,t+1}}^{-1}/\gamma_{V_{i,t}}^{-1} + \gamma_{V_{i,t}}^{-1}.
\]

These extrinsic precisions and means are updated in lines 13-14 of Algorithm 2. Given this extrinsic information, the denoising functions, \(g_u(\cdot, \cdot)\) and \(g_v(\cdot, \cdot)\), used to estimate \(u_i^{+}\) and \(v_i^{+}\), along with their respective divergences, \(g_u'(\cdot, \cdot)\) and \(g_v'(\cdot, \cdot)\), in lines 18-22 of Algorithm 2 are given by:

\[
p_y(u_i^{+}) = \int u_p(u_i^{+}) N(u; \widehat{u}, \beta^{-1} \gamma U_i^{1}) \, du, \\
p_y(u_i^{+}) = \int v_p(v_i^{+}) N(v; \widehat{v}, \beta^{-1} \gamma V_i^{1}) \, dv.
\]

Finally, the extrinsic precisions and means for the messages (3) and (4) are updated analogously to (33-36) in lines 22-25 of Algorithm 2.

B. From Bi-VAMP to BiG-VAMP

In this section, we extend the Bi-VAMP algorithm introduced in Section III-A to the generalized bilinear model given in (1) so as to complete the derivation of BiG-VAMP. To that end, we introduce the intermediate random matrix, \(Z \triangleq UV^\top\), whose \(ij\)th entry is given by \(z_{ij} \triangleq u_i \cdot v_j\). We again resort to the expectation propagation (aka turbo) principle to approximate the posterior messages (5) and (6) in Fig. 3 by Gaussian distributions, \(N(z_{ij}; \hat{z}_{ij}, \beta^{-1} \gamma_{Z_{ij},t}^{-1})\) and \(N(z_{ij}; \hat{z}_{ij}, \beta^{-1} \gamma_{Z_{ij},t}^{-1})\), respectively, whose means and variances are calculated in the sequel. To start with, by defining \(\tilde{z}_{ij,p} \triangleq u_i - v_j\), the posterior mean and variance, \(\tilde{z}_{ij,p}^{-1}\) and \(\gamma_{\tilde{z}_{ij,p},t}^{-1}\), under the scalar likelihood, \(p_{y_{ij}}(\cdot|y_{ij})\), are obtained as follows:

\[
\gamma_{\tilde{z}_{ij,p},t}^{-1} = \frac{\partial g_z(y_{ij}, \tilde{z}_{ij,p}, t, \gamma_{Z_{ij},t}^{-1})}{\partial \tilde{z}_{ij,p,t+1}},
\]

\[
\gamma_{\tilde{z}_{ij,p},t}^{-1} = \frac{\partial g_z(y_{ij}, \tilde{z}_{ij,p}, t, \gamma_{Z_{ij},t}^{-1})}{\partial \tilde{z}_{ij,p,t+1}}.
\]
Now, we detail the derivation of the scalar extrinsic message \( \delta(\mathbf{u}; \mathbf{v}) \) which is approximated by the Gaussian density
\[
\mathcal{N}(\mathbf{j}_i; \mathbf{y}_i, \mathbf{t}+1, \beta^{-1} \gamma^{-1}_z z_{ij}^{-1})
\]
with a common variance for all nodes. We start with the derivation of the individual variance,
\[ \beta^{-1} \gamma^{-1}_z z_{ij}^{-1} \] of \( z_{ij} \):
\[
\beta^{-1} \gamma^{-1}_z z_{ij}^{-1} = E \left[ (\mathbf{u}_i - \mathbf{v}_j) (\mathbf{u}_i - \mathbf{v}_j) \right] \tag{46}
\]
In (46), the expectation is taken with respect to the densities on \( \mathbf{u}_i \) and \( \mathbf{v}_j \) while assuming them to be independent. Those densities are given by messages (3) and (3), at iteration at iteration \( t \), whose first- and second-order statistics were already evaluated in (16)-(19). Therefore, it follows from (46) that:
\[
\beta^{-1} \gamma^{-1}_z z_{ij}^{-1} = E \left[ (\mathbf{u}_i - \mathbf{v}_j) (\mathbf{u}_i - \mathbf{v}_j) \right] = E \left[ (\mathbf{u}_i - \mathbf{v}_j) (\mathbf{u}_i - \mathbf{v}_j) \right] = \text{Tr} \left[ \left( \mathbf{u}_i - \mathbf{v}_j \right) \left( \mathbf{u}_i - \mathbf{v}_j \right) \right] = \left( \mathbf{u}_i - \mathbf{v}_j \right) \left( \mathbf{u}_i - \mathbf{v}_j \right)
\]
with \( g(x, y, \gamma^{-1}) \) being the following scalar denoising function:
\[
g(x, y, \gamma^{-1}) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi \gamma^{-1}}} e^{-\frac{1}{2\gamma^{-1}} (x - y)^2} \, dx
\]
\[
\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi \gamma^{-1}}} e^{-\frac{1}{2\gamma^{-1}} (x - y)^2} \, dx
\]
\[
\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi \gamma^{-1}}} e^{-\frac{1}{2\gamma^{-1}} (x - y)^2} \, dx
\]
Using the covariance identity, $\text{cov}(x, x) = \mathbb{E}[xx^T] - \mathbb{E}[x] \mathbb{E}[x]^T$, for any random vector $x$, (47) is equivalent to:

$$
\beta^{-1} \gamma^{-1}_{ij,t+1} = \text{Tr} \left( \left[ \text{cov}(u_i, u_i) + \hat{u}_{i-\to(i),t+1} u_{i-\to(i),t+1}^T \right] \times \left[ \text{cov}(v_j, v_j) + \hat{v}_{j-\to(j),t+1} v_{j-\to(j),t+1}^T \right] \right) - \left( \hat{u}_{i-\to(i),t+1} v_{j-\to(j),t+1}^T \right)^2. \quad (48)
$$

By recalling the fact that $\text{cov}(u_i, u_i) \triangleq \beta^{-1} R_u, \text{cov}(v_j, v_j) \triangleq \beta^{-1} R_v$, and with common $R_u$ and $R_v$ from line 8 and 10 in Algorithm 2, it follows that:

$$
\gamma^{-1}_{ij,t+1} = \text{Tr} \left( \beta^{-1} R_u u_{i-\to(i),t+1} v_{j-\to(j),t+1}^T \right) + \beta^{-1} R_v v_{j-\to(j),t+1} u_{i-\to(i),t+1}^T + R_{uv}_{i-\to(i),t+1} v_{j-\to(j),t+1}^T u_{i-\to(i),t+1}, \quad (49)
$$

In (49), we further replace $\hat{u}_{i-\to(i),t+1}$ and $\hat{v}_{j-\to(j),t+1}$ by their broadcast versions $\hat{u}_{i,p,t+1}$ and $\hat{v}_{j,p,t+1}$, respectively, while incurring a negligible error after ignoring terms of vanishing order as $M$ and $N$ grow large:

$$
\gamma^{-1}_{ij,t+1} = \text{Tr} \left( \beta^{-1} R_u u_{i-\to(i),t+1} v_{j-\to(j),t+1}^T \right) + \beta^{-1} R_v v_{j-\to(j),t+1} u_{i-\to(i),t+1}^T + R_{uv}_{i-\to(i),t+1} v_{j-\to(j),t+1}^T u_{i-\to(i),t+1}. \quad (50)
$$

As usually done in approximate message passing practices, we combine the individual variances, $\gamma^{-1}_{ij,t+1}$, into one common variance, $\gamma^{-1}_{z_{-t+1}}$, for all nodes:

$$
\gamma^{-1}_{z_{-t+1}} = \frac{1}{MN} \sum_{i=1}^N \sum_{j=1}^M \gamma^{-1}_{ij,t+1} \approx \text{Tr} \left( \beta^{-1} R_u u_{i-\to(i),t+1} v_{j-\to(j),t+1}^T \right) + \beta^{-1} R_v v_{j-\to(j),t+1} u_{i-\to(i),t+1}^T + R_{uv}_{i-\to(i),t+1} v_{j-\to(j),t+1}^T u_{i-\to(i),t+1}, \quad (51)
$$

Moreover, the posterior mean, $\hat{z}_{ij,t+1}$, is evaluated as follows:

$$
\hat{z}_{ij,t+1} = \gamma^{-1}_{z_{-t+1}} \left( \gamma_{z_{-t+1}} \hat{z}_{ij,e,t} + \gamma_{z_{-t+1}} \hat{u}_{i-\to(i),t+1} \hat{v}_{j-\to(j),t+1} \right). \quad (52)
$$

Recall here the Osanger correction terms in (29a) and (29b) which lead to the following approximations of $\hat{u}_{i-\to(i),t+1}$ and $\hat{v}_{j-\to(j),t+1}$:

$$
\hat{u}_{i-\to(i),t+1} \approx \hat{u}_{i,p,t+1} - \gamma_{z_{-t+1}} \hat{z}_{ij,e,t} R_u u_{i-\to(i),t+1} \hat{v}_{j-\to(j),t+1}, \quad (53a)
$$

$$
\hat{v}_{j-\to(j),t+1} \approx \hat{v}_{j,p,t+1} - \gamma_{z_{-t+1}} \hat{z}_{ij,e,t} R_v v_{j-\to(j),t+1} \hat{u}_{i-\to(i),t+1}. \quad (53b)
$$

These approximated messages are injected back into (52), thereby yielding the approximate posterior mean in (54) on the top of the next page. The approximation in (54a) is a result of dropping the last term which has a vanishing order as $M \to \infty$. The approximation in (54d) follows from the observation that one makes an error of vanishing order (as both $M$ and $N$ grow large) due to the fact that $\hat{u}_{i,p,t+1} \hat{u}_{i,p,t+1} \approx \hat{u}_{i,p,t+1} \hat{v}_{j,p,t+1}$ and $\hat{v}_{j,p,t+1} \hat{v}_{j,p,t+1} \approx \hat{v}_{j,p,t+1} \hat{v}_{j,p,t+1}$. Then using (50) in (54d) leads to (54e) in which we further replace $\hat{z}_{ij,e,t}$ by $\gamma_{z_{-t+1}}$ thereby yielding:

$$
\hat{z}_{ij,p,t+1} = \frac{\gamma_{z_{-t+1}}}{\gamma_{z_{-t+1}}} \left( \hat{u}_{i,p,t+1} \hat{v}_{j,p,t+1} \right) + \beta^{-1} \gamma_{z_{-t+1}} \gamma_{z_{-t+1}} \text{Tr}(R_{uv}_{i-\to(i),t+1} R_{uv}_{j-\to(j),t+1} \hat{v}_{j,p,t+1} \hat{u}_{i,p,t+1}). \quad (55)
$$

After plugging the expression of $\gamma_{z_{-t+1}}$ as obtained from (51), that is $\gamma_{z_{-t+1}} = \gamma_{z_{-t+1}} + \gamma_{z_{-t+1}}$, (55) becomes:

$$
\hat{z}_{ij,p,t+1} = \frac{\gamma_{z_{-t+1}}}{\gamma_{z_{-t+1}}} \left( \hat{u}_{i,p,t+1} \hat{v}_{j,p,t+1} \right) + \beta^{-1} \gamma_{z_{-t+1}} \gamma_{z_{-t+1}} \text{Tr}(R_{uv}_{i-\to(i),t+1} R_{uv}_{j-\to(j),t+1} \hat{v}_{j,p,t+1} \hat{u}_{i,p,t+1}). \quad (56)
$$

Finally, we assume that $\gamma_{z_{-t+1}} \gg \gamma_{z_{-t+1}}$ which follows from the observation that the information on $z_{ij} = u_i v_j$ that is brought by the strong structure on both $U$ and $V$ overwhelms the information brought by the observation $Y$. This leads to:

$$
\hat{z}_{ij,p,t+1} = \frac{\gamma_{z_{-t+1}}}{\gamma_{z_{-t+1}}} \left( \hat{u}_{i,p,t+1} \hat{v}_{j,p,t+1} \right) + \beta^{-1} \gamma_{z_{-t+1}} \gamma_{z_{-t+1}} \text{Tr}(R_{uv}_{i-\to(i),t+1} R_{uv}_{j-\to(j),t+1} \hat{v}_{j,p,t+1} \hat{u}_{i,p,t+1}). \quad (57)
$$

In summary, (57) and (51) are the element-wise expressions of (58a) and (58b), respectively:

$$
\hat{z}_{p,t+1} = \hat{U}_{e,t+1} \hat{V}_{e,t+1} + \frac{\gamma_{z_{-t+1}}}{\beta_{Z_{-t+1}}} \text{Tr}(R_{uv_{-t+1} R_{uv_{-t+1}} \hat{v}_{j,p,t+1} \hat{u}_{i,p,t+1}), \quad (58a)
$$

$$
\gamma_{z_{-t+1}} = \gamma_{z_{-t+1}} + MN \text{Tr}(R_{uv_{-t+1} R_{uv_{-t+1}} \hat{v}_{j,p,t+1} \hat{u}_{i,p,t+1}), \quad (58b)
$$

which correspond to lines (26)(27) in Algorithm 2. The extrinsic values, $\hat{z}_{ij,e,t+1}$ and $\gamma^{-1}_{z_{-t+1}}$, for message 5 in Fig. 5 are then easily evaluated. Finally, the extrinsic mean and variance, $\hat{z}_{ij,e,t+1}$ and $\gamma^{-1}_{z_{-t+1}}$, of message 6 can be calculated from the posterior mean, $\gamma^{-1}_{z_{-t+1}}$, and common precision, $\gamma_{z_{-t+1}}$, as specified in lines (51)(52) of Algorithm 2.

IV. STATE EVOLUTION

Our main goal is to understand the behavior of the proposed BiG-VAMP algorithm in the asymptotic regime for a certain class of $U$ and $V$ matrices, i.e., when they both have zero mean i.i.d. priors. For simplicity, we will focus on MMSE estimation, i.e., $\beta = 1$. In our case, the asymptotic regime refers to the case where
\[ \tilde{z}_{i,j,p,t+1} = \gamma_{Z_{p,t+1}}^{-1} \left[ \gamma_{Z_{p,t}}^{-1} \tilde{z}_{i,j,e,t} + \gamma_{z_{i,j,t}}^{-1} \left( \tilde{u}_{i,p,t+1} - \gamma_{Z_{p,t}}^{-1} \tilde{z}_{i,j,e,t} R_{U_{p,t+1}} \tilde{v}_{j,p,t} \right) \right] \]

...continuing the mathematical derivation...
\( \mathcal{E}_u^- = \lim_{r \to \infty} \frac{1}{r} \text{Tr} (R_{U_r}) = \lim_{M, N \to \infty} \frac{1}{r} \text{Tr} \left( \left[ \gamma_{U_r} I + \frac{\gamma_w}{\sqrt{MN}} \hat{V}_p^- \hat{V}_p \right] \right)^{-1} \), \( \mathcal{E}_v^- = \lim_{r \to \infty} \frac{1}{r} \text{Tr} (R_{V_r}) = \lim_{M, N \to \infty} \frac{1}{r} \text{Tr} \left( \left[ \gamma_{V_r} I + \frac{\gamma_w}{\sqrt{MN}} \hat{U}_p^- \hat{U}_p \right] \right)^{-1} \). \( \gamma_{U_r} \) and \( \gamma_{V_r} \) by \( \bar{\gamma}_{U_r} \) and \( \bar{\gamma}_{V_r} \), respectively. After using these approximations in (61) and (62), it follows that:

\( \mathcal{E}_u^- (\bar{\gamma}_{U_r}) = \lim_{r \to \infty} \frac{1}{r} \text{Tr} \left( \left[ \bar{\gamma}_{U_r} I + \frac{\gamma_w}{\sqrt{MN}} \hat{V}_p^- \hat{V}_p \right] \right)^{-1} \), \( \mathcal{E}_v^- (\bar{\gamma}_{V_r}) = \lim_{r \to \infty} \frac{1}{r} \text{Tr} \left( \left[ \bar{\gamma}_{V_r} I + \frac{\gamma_w}{\sqrt{MN}} \hat{U}_p^- \hat{U}_p \right] \right)^{-1} \), \( \bar{\gamma}_{U_r} = \gamma_{U_r} + 1 - \sqrt{\frac{\beta_u}{\beta_v}} \gamma_{V_r} \), \( \bar{\gamma}_{V_r} = \gamma_{V_r} + 1 - \sqrt{\frac{\beta_v}{\beta_u}} \gamma_{U_r} \).

To find the limits in (65) and (66), we define the following two matrices:

\( H_u = \frac{1}{ \sqrt{(\sigma_u^2 - \bar{\gamma}_{U_r}^2)} N} \hat{U}_p \), \( H_v = \frac{1}{ \sqrt{(\sigma_v^2 - \bar{\gamma}_{V_r}^2)} M} \hat{V}_p \). \( \bar{\gamma}_{U_r} \) and \( \bar{\gamma}_{V_r} \) are i.i.d. with zero mean and variance \( \frac{1}{N} \) (resp., \( \frac{1}{M} \)). For ease of notation, we also define the two quantities:

\( \alpha_u \triangleq \frac{\sqrt{\beta_u} \gamma_w (\sigma_u^2 - \bar{\gamma}_{U_r}^2)}{\beta_u \bar{\gamma}_{V_r}} \), \( \alpha_v \triangleq \frac{\sqrt{\beta_v} \gamma_w (\sigma_v^2 - \bar{\gamma}_{V_r}^2)}{\beta_v \bar{\gamma}_{U_r}} \).

Now, using a well-known result in random matrix theory (cf. eq. (1.16) in [23]) — we show that:

\( \mathcal{E}_u^- (\bar{\gamma}_{U_r}) = \lim_{r \to \infty} \frac{1}{r} \text{Tr} \left( \left[ \bar{\gamma}_{U_r} I + \frac{\gamma_w}{\sqrt{MN}} \hat{V}_p^- \hat{V}_p \right] \right)^{-1} \), \( \mathcal{E}_v^- (\bar{\gamma}_{V_r}) = \lim_{r \to \infty} \frac{1}{r} \text{Tr} \left( \left[ \bar{\gamma}_{V_r} I + \frac{\gamma_w}{\sqrt{MN}} \hat{U}_p^- \hat{U}_p \right] \right)^{-1} \), \( \bar{\gamma}_{U_r} \) and \( \bar{\gamma}_{V_r} \) is defined as:

\[ \mathcal{F}(x, z) = \left( \sqrt{x(1 + \sqrt{2})^2 + 1} - \sqrt{x(1 - \sqrt{2})^2 + 1} \right)^2. \]

Similarly we show that:

\( \mathcal{E}_u^- (\bar{\gamma}_{U_r}) = \lim_{r \to \infty} \frac{1}{r} \text{Tr} \left( \left[ \bar{\gamma}_{U_r} I + \frac{\gamma_w}{\sqrt{MN}} \hat{U}_p^- \hat{U}_p \right] \right)^{-1} \), \( \mathcal{E}_v^- (\bar{\gamma}_{V_r}) = \lim_{r \to \infty} \frac{1}{r} \text{Tr} \left( \left[ \bar{\gamma}_{V_r} I + \frac{\gamma_w}{\sqrt{MN}} \hat{U}_p^- \hat{U}_p \right] \right)^{-1} \), \( \bar{\gamma}_{U_r} \) and \( \bar{\gamma}_{V_r} \) is defined as:

\[ \mathcal{F}(x, z) = \left( \sqrt{x(1 + \sqrt{2})^2 + 1} - \sqrt{x(1 - \sqrt{2})^2 + 1} \right)^2. \]

Recall here that as \( r, N, M \to +\infty \) we have \( \bar{\gamma}_{U_r} \to \beta_u \) and \( \bar{\gamma}_{V_r} \to \beta_v \). Now, the output variances of the MMSE denoisers of \( \hat{U} \) and \( \hat{V} \) matrices are obtained from lines \( 20 \) and \( 21 \) of Algorithm \( 2 \) as follows:

\( \mathcal{E}_u^+ (\gamma_{U_r}) \triangleq \frac{1}{N \gamma_{U_r}} \sum_{i=1}^{N} (g_i' (\bar{u}_{r, i}, \gamma_{U_r}^2)), \) \( \mathcal{E}_v^+ (\gamma_{V_r}) \triangleq \frac{1}{M \gamma_{V_r}} \sum_{j=1}^{M} (g_j' (\bar{v}_{r, j}, \gamma_{V_r}^2))). \)

In the large system limits, the empirical averages involved in (74) and (75) can be approximated by the following statistical averages in which we use the fact that \( \gamma_{U_r} \to \bar{\gamma}_{U_r} \) and \( \gamma_{V_r} \to \bar{\gamma}_{V_r} \) as \( M \) and \( N \) grow large:

\( \mathcal{E}_u^+ (\bar{\gamma}_{U_r}) \triangleq \frac{1}{N \bar{\gamma}_{U_r}} \mathbb{E} \left[ g_i' (\bar{u}_{r, \bar{U}_r}^2, \gamma_{U_r}^2) \right] p_u(u) p_{\bar{u}_{U_r}} |u (\bar{u}_{r, \bar{U}_r}) \),

\( \mathcal{E}_v^+ (\bar{\gamma}_{V_r}) \triangleq \frac{1}{M \bar{\gamma}_{V_r}} \mathbb{E} \left[ g_j' (\bar{v}_{r, \bar{V}_r}^2, \gamma_{V_r}^2) \right] p_v(v) p_{\bar{v}_{V_r}} |v (\bar{v}_{r, \bar{V}_r}) \).
\[ E_{v+}(\tilde{\gamma}_V^-) = \frac{1}{\tilde{\gamma}_V^-} E \left[ g_v(\tilde{\gamma}_e^-; \tilde{\gamma}_V^-) p_v(v) p_{\tilde{\gamma}_e^-} [\tilde{p}(\tilde{\gamma}_e^- | v)] \right] \]  

(77)

In (76) and (77), \( p_{\tilde{\gamma}_e^-}[\tilde{u}_e^- | u] \) and \( p_{\tilde{\gamma}_e^-}[\tilde{v}_e^- | v] \) correspond to the scalar models \( u_e^- = u + w_u \) and \( v_e^- = v + w_v \), where under the matched conditions, we have \( w_u \sim N(0, \tilde{\gamma}_U^-) \) and \( w_v \sim N(0, \tilde{\gamma}_V^-) \). Under all the aforementioned assumptions and matched conditions, we can now describe in Algorithm 3 our main result, which is the SE recursion equations for Bi-VAMP:

**Algorithm 3 Bi-VAMP State Evolution**

- **Require**: Noise precision \( \gamma_u \); set \( \beta = 1 \); \( \beta_u \) and \( \beta_v \); number of iterations \( T_{\text{max}} \)
- **Initialization**: extrinsic precisions \( \tilde{\gamma}_U^+, t ^{-1} \), \( \tilde{\gamma}_V^+, t ^{-1} \)

1. for \( t = 1, \ldots, T_{\text{max}} \) do
   2. \( \tilde{\gamma}_U^+, t + 1 = \tilde{\gamma}_U^+, t - \frac{\sqrt{\beta u \tilde{\gamma}_U^-} \tilde{\gamma}_V^+, t}{\sqrt{\beta u \tilde{\gamma}_U^+}} \left( \sqrt{\beta u \tilde{\gamma}_U^-} \tilde{\gamma}_V^+, t + 1 \right) \)
   3. \( \tilde{\gamma}_V^+, t + 1 = \tilde{\gamma}_V^+, t - \frac{\sqrt{\beta u \tilde{\gamma}_V^-} \tilde{\gamma}_U^+, t}{\sqrt{\beta u \tilde{\gamma}_V^+}} \left( \sqrt{\beta u \tilde{\gamma}_V^-} \tilde{\gamma}_U^+, t + 1 \right) \)
   4. \( \tilde{\gamma}_U^-, t + 1 = \frac{1}{\tilde{\gamma}_U^+, t + 1} \)
   5. \( \tilde{\gamma}_V^-, t + 1 = \frac{1}{\tilde{\gamma}_V^+, t + 1} \)
   6. \( \tilde{\gamma}_U^+, t + 1 = \tilde{\gamma}_U^+, t + 1 \)
   7. \( \tilde{\gamma}_V^+, t + 1 = \tilde{\gamma}_V^+, t + 1 \)
   8. \( \tilde{\gamma}_U^+, t + 1 = \tilde{\gamma}_U^+, t + 1 \)
   9. \( \tilde{\gamma}_V^+, t + 1 = \tilde{\gamma}_V^+, t + 1 \)
   10. \( \tilde{\gamma}_U^-, t + 1 = \tilde{\gamma}_U^-, t + 1 \)
   11. \( \tilde{\gamma}_V^-, t + 1 = \tilde{\gamma}_V^-, t + 1 \)
12. end for
13. Return \( \tilde{\gamma}_U^+, T_{\text{max}} + 1; \tilde{\gamma}_V^+, T_{\text{max}} + 1 \)

Note here that at convergence, one must have equality between the posterior variances of the same variable:

\[ E_{u+}(\tilde{\gamma}_U^+, \infty) = \tilde{\gamma}_U^-, \infty \]
\[ E_{v+}(\tilde{\gamma}_V^+, \infty) = \tilde{\gamma}_V^-, \infty \]
\[ E_{u-}(\tilde{\gamma}_U^-, \infty) = \tilde{\gamma}_U^+, \infty \]
\[ E_{v-}(\tilde{\gamma}_V^-, \infty) = \tilde{\gamma}_V^+, \infty \]

(78)

**B. Extension to the Generalized Bilinear Model: State Evolution of BiG-VAMP**

In this case, the observation matrix, \( Y \), is obtained from the generalized bilinear model in (1). To account for the residual error of the output denoiser (cf. Fig. 4), all the previous SE equations summarized in Algorithm 3 remain the same except for replacing the noise variance, \( \gamma_u^- \), by the extrinsic error variance, \( \gamma_u^- \), whose SE update equation will be characterized in the following.

Again, to maintain the same energy per each entry in the matrix \( UV \), as we grow \( M \) and \( N \), we redefine \( Z \) as \( Z = \frac{1}{\sqrt{MN}} UV \).

Recall form line 27 in Algorithm 2 that the posterior variance, \( E_{\gamma^-}(\gamma_{Z^-}^+ \infty) = \gamma_{Z^-}^+ \), of each \( z_{ij} \) estimate provided by the Bi-MMSE module is given by (after taking into account the effect of the above scaling by \( \frac{1}{\sqrt{MN}} \)):

\begin{equation}
E_{\gamma^-}(\gamma_{Z^-}^+) = \gamma_{Z^-}^+ + \sqrt{MN} \left[ \text{Tr} \left( \frac{1}{\beta} R_{U^+} R_{V^+}^T + \frac{1}{M} R_{U^+} V_{\tilde{p}}^T V_{\tilde{p}} + \frac{1}{N} R_{V^+} U_{\tilde{p}}^T U_{\tilde{p}} \right) \right]^{-1}.
\end{equation}

(79)

For large enough \( M \) and \( N \), by plugging (63a), (63b), and (69) in (79), it follows that:

\begin{equation}
E_{\gamma^+}(\gamma_{Z^+}^-) = \frac{1}{\gamma_{Z^+}^-} = \frac{1}{MN / \gamma_{Z^+}^-} \sum_{i=1}^N \sum_{j=1}^M \hat{g}_i(y_{ij}, \hat{z}_{ij}; \tilde{\gamma}_{Z^+}^-) \gamma_{Z^+}^-.
\end{equation}

(81)

In the large system limits, the empirical average involved in (81) can be approximated by the following statistical average in which we use the fact that \( \gamma_{Z^+}^- \rightarrow \gamma_{Z^+}^- \) as \( M \) and \( N \) grow large:

\begin{equation}
E_{\gamma^+}(\gamma_{Z^+}^-) = \frac{1}{\gamma_{Z^+}^-} = \frac{1}{MN / \gamma_{Z^+}^-} \sum_{i=1}^N \sum_{j=1}^M \hat{g}_i(y_{ij}, \hat{z}_{ij}; \tilde{\gamma}_{Z^+}^-) \gamma_{Z^+}^-.
\end{equation}

(82)

In (82), \( \hat{p}_z(z | \hat{z}_{\gamma^-}^-) \) corresponds to the scalar model \( \hat{z}_{\gamma^-}^- = z + w_z \) where under the matched conditions, we have \( w_z \sim N(0, \tilde{\gamma}_z^-) \). Moreover, since we have \( z = \frac{1}{\sqrt{MN}} U^T v \), then owing to the central limit theorem we have \( \hat{p}_z(z) = N(z; 0, \sqrt{\beta u \tilde{\gamma}_U^-} \sigma_z^2 / \tilde{\gamma}_V^- \). Under all the assumptions and matched conditions that we stated above, we can now describe in Algorithm 3 our main result, which is the SE recursion for BiG-VAMP. Note that at convergence, on top of the equalities in (78), we also have:

\begin{equation}
E_{\gamma^+}(\gamma_{Z^+}^-) = \frac{1}{\gamma_{Z^+}^-} = \frac{1}{MN / \gamma_{Z^+}^-} \sum_{i=1}^N \sum_{j=1}^M \hat{g}_i(y_{ij}, \hat{z}_{ij}; \tilde{\gamma}_{Z^+}^-) \gamma_{Z^+}^-.
\end{equation}

(83)

**V. Simulation Results**

In this section, we assess the performance behavior of the proposed BiG-VAMP algorithm and benchmark it against BiG-AMP [21], BiD-VAMP [23] and LowRAMP [20] algorithms for different applications, namely:

- matrix factorization,
- dictionary learning,
- matrix completion.
Algorithm 4 BiG-VAMP State Evolution

Require : set $\beta = 1; \beta_u$ and $\beta_v$; number of iterations $T_{\text{max}}$.
Initialization : extrinsic precisions $\tau Z^+, \tilde{\tau} U^+, \tilde{\tau} V^+, \gamma U^{-}, \gamma V^-.$

1: for $t = 1, \ldots, T_{\text{max}}$ do
    2: $\tilde{\tau} U^+, t+1 = \tau U^+, t + \frac{1}{\beta_u} \sqrt{\tau U^+, t} \tilde{Z}_t^{-1} \gamma V^-, t \tilde{\tau} V^-, t \\
    3: \quad - \sqrt{\tau U^+, t} \tilde{Z}_t^{-1} \gamma V^-, t \tilde{\tau} V^-, t \\
    4: \tilde{\tau} V^-, t+1 = \tau V^-, t + \frac{1}{\beta_v} \sqrt{\tau V^-, t} \tilde{Z}_t^{-1} \gamma U^+, t \tilde{\tau} U^+, t \\
    5: \quad - \sqrt{\tau V^-, t} \tilde{Z}_t^{-1} \gamma U^+, t \tilde{\tau} U^+, t \\
    \triangleright$ compute the effective inverse noise variance for the Bi-LMMSE block
    6: $\tilde{\tau} Z^-, t+1 = \tilde{\tau} Z^-, t + \frac{1}{\beta_v} \sqrt{\tau Z^-, t} \tilde{Z}_t^{-1} \gamma V^-, t \tilde{\tau} V^-, t \\
    7: \quad - \sqrt{\tau Z^-, t} \tilde{Z}_t^{-1} \gamma V^-, t \tilde{\tau} V^-, t \\
    \triangleright$ compute the analytical posterior and extrinsic precision of $u$
    8: $\tilde{\tau} U^+, t+1 = \tilde{\tau} U^+, t + \frac{1}{\beta_u} \sqrt{\tau U^+, t} \tilde{Z}_t^{-1} \gamma V^-, t \tilde{\tau} V^-, t \\
    9: \quad - \sqrt{\tau U^+, t} \tilde{Z}_t^{-1} \gamma V^-, t \tilde{\tau} V^-, t \\
    \triangleright$ compute the analytical posterior and extrinsic precision of $v$
    10: $\tilde{\tau} V^-, t+1 = \tilde{\tau} V^-, t + \frac{1}{\beta_v} \sqrt{\tau V^-, t} \tilde{Z}_t^{-1} \gamma U^+, t \tilde{\tau} U^+, t \\
    11: \quad - \sqrt{\tau V^-, t} \tilde{Z}_t^{-1} \gamma U^+, t \tilde{\tau} U^+, t \\
    \triangleright$ compute the analytical posterior and extrinsic precision of $z$
    12: $\tilde{\tau} Z^+, t+1 = \tilde{\tau} Z^+, t + \frac{1}{\beta_v} \sqrt{\tau Z^+, t} \tilde{Z}_t^{-1} \gamma V^-, t \tilde{\tau} V^-, t \\
    13: \quad - \sqrt{\tau Z^+, t} \tilde{Z}_t^{-1} \gamma V^-, t \tilde{\tau} V^-, t \\
    \triangleright$ compute the analytical posterior and extrinsic precision of $u$
    14: $\tilde{\tau} U^+, t+1 = \tilde{\tau} U^+, t + \frac{1}{\beta_u} \sqrt{\tau U^+, t} \tilde{Z}_t^{-1} \gamma V^-, t \tilde{\tau} V^-, t \\
    15: \quad - \sqrt{\tau U^+, t} \tilde{Z}_t^{-1} \gamma V^-, t \tilde{\tau} V^-, t \\
    \triangleright$ compute the analytical posterior and extrinsic precision of $v$
    16: \end for
17: Return $\gamma U^+, T_{\text{max}} + 1; \tilde{\gamma} V^+, T_{\text{max}} + 1; \tilde{\gamma} Z^+, T_{\text{max}} + 1$

In all simulations, we set $T_{\text{max}} = 1000$ and the precision tolerance to $\xi = 10^{-6}$ and we perform $N_{\text{MC}} = 100$ Monte-Carlo trials for different values of the SNR:

$$\text{SNR} = 10 \log_{10} \left( \frac{\|Z\|^2_F}{\|W\|^2_F} \right),$$

where $\|.\|_F$ is the Frobenius norm. We also use the normalized root MSE (NRMSE) as performance measure which defined as follows:

$$\text{NRMSE} = \frac{1}{N_{\text{MC}}} \sum_{t=1}^{N_{\text{MC}}} \frac{\|Z_t - \hat{Z}_t\|_F}{\|Z_t\|_F},$$

where $Z_t$ is the realization of $Z$ and $\hat{Z}_t$ is its reconstruction during the $t$th Monte-Carlo trial. As per BiG-VAMP’s initialization setting, all the initial means and covariances were set to the all-zero vector and the identity matrix, respectively. The results disclosed in the sequel demonstrate that BiG-VAMP yields considerable improvements in reconstruction performance and robustness as compared to state-of-the-art algorithms, especially in presence of discrete priors on either $U$ or $V$.

A. Noisy dictionary learning

Here, we apply the proposed BiG-VAMP algorithm to the well-known dictionary learning problem wherein the goal is to find, from a noisy observation $Y$, a dictionary matrix $U$ and a sparse matrix $V$. For that purpose, we use Gaussian and Bernoulli-Gaussian priors on $U$ and $V$ matrices, respectively. We depict the NRMSE on the estimated $\hat{Z}$ in Fig. 6 wherein we benchmark the proposed algorithm against BiG-AMP and BAd-VAMP. Note that BAd-VAMP was simulated with its defaults parameters from [23], i.e., $\tau_1, \tau_2 = 0$ and $\gamma_{\text{min}} = 10^{-6}$.

In Fig. 6, it is seen that BiG-AMP and BAd-VAMP exhibit the same performance as BiG-VAMP which is hardly surprising since all algorithms are optimally exploiting the considered priors on $U$ and $V$ and do not suffer from any convergence issues.

Next, we consider a binary prior on the unknown dictionary $U$. Note that in this case the underlying dictionary learning problem is also known as the $Z/2$ synchronization problem in the mathematical literature [28] or blind detection problem in the communication literature [29]. In this context, we again compare BiG-VAMP to BiG-AMP and we consider both low rank (i.e., $r = 5$) and moderately high rank (i.e., $r = 25$) structures as illustrated in Figs. 7 and 8 respectively.

We do not include BAd-VAMP in the subsequent simulations since it is not designed to handle non-Gaussian priors on $U$ This is in fact due to the inherent limitation imposed by the combinatorial maximization step in the EM algorithm that BAd-VAMP uses to update $U$ at every iteration.

Figs. 7 and 8 show order of magnitude difference in the NRMSE performance of BiG-VAMP and BiG-AMP. While BiG-VAMP outperforms BiG-AMP under both low-rank and high-rank structures, the gap between the two algorithms is not inherently related to the rank value, but rather to the inability of BiG-AMP to converge under discrete priors. Moreover, it is seen from both figures that the empirical NRMSE of BiG-VAMP is

$^7$Note here that the Gaussian prior on $U$ was incorporated during the maximization step of the EM algorithm inside BAd-VAMP. In this special case, BAd-VAMP is actually performing MAP estimation of $U$ (instead of maximum likelihood estimation).
Fig. 7: NRMSE of BiG-VAMP and BiG-AMP vs. the SNR for the dictionary learning problem: binary prior on $U$, Bernoulli-Gaussian prior on $V$ with a sparsity level of 95%, $N = 100$, $M = 100$ and $r = 5$.

Fig. 8: NRMSE of BiG-VAMP and BiG-AMP vs. the SNR for the dictionary learning problem: binary prior on $U$, Bernoulli-Gaussian prior on $V$ with a sparsity level of 95%, $N = 500$, $M = 500$ and $r = 25$.

Fig. 9: NRMSE of BiG-VAMP and BiG-AMP vs. the SNR for the matrix factorization problem: binary prior on $U$, Gaussian prior on $V$, $N = 1000$, $M = 200$ and $r = 30$.

Fig. 10: NRMSE of BiG-VAMP and its SE for the matrix factorization problem at SNR = 10 dB: binary prior on $U$, Gaussian prior on $V$, $N = 1000$, $M = 500$ and $r = 10$.

accurately predicted by the analytical state evolution recursion thereby corroborating the theoretical analysis we conducted in Section IV.

B. Matrix factorization

In this case, we compare BiG-VAMP to BIG-AMP [26] for the case where $U$ is a binary matrix and $V$ is Gaussian-distributed. Fig. 9 depicts the NRMSE of both algorithms and reveals that BiG-AMP’s performance again deteriorates considerably in presence of a discrete prior either on $U$ or $V$ (due to the symmetry of the problem). BiG-VAMP, however, finds an accurate factorisation of $Z$ over the entire SNR range and its empirical NRMSE is again theoretically predicted by the established state evolution analysis. This endows the proposed algorithm with offline design guidelines when applied to different engineering problems in practice.

A close inspection of the NRMSE time evolution, as shown in Fig. 10 illustrates qualitatively a transiently chaotic trajectory that is similar to those of fluid parcels in turbulent flows studied in [3]. Such a chaotic behaviour may be a generic feature of algorithms searching for solutions in hard optimization problems with applications as diverse as protein folding and Sudoku [3].

C. Matrix completion

Unlike the previous two applications (i.e., matrix factorization and dictionary learning) wherein the observation model was linear, we now turn our attention to a popular generalized bilinear recovery problem, namely the matrix completion problem. In this context, Fig. 11 compares BiG-VAMP to BIG-AMP under the nonlinear observation model in (2) by taking $\phi(\cdot)$ to be a random selection with a rate of 20%. There, it is also seen that BiG-VAMP outperforms by far BiG-AMP which is in principle able to deal with any non-linearity in the observation model. But its deficiency here stems from the fact that it diverges for the considered challenging problem.

For the case where the only problem structure available at hand is the low rank approximation, we also benchmark BiG-VAMP against LowRAMP [20] whose MATLAB code is publicly available from [30]. The results are depicted in Fig. 11 and as
seen there LowRAMP is not able to correctly recover $U$ and $V$ while BiG-VAMP exhibits the same performance as BiG-VAMP. Under the considered non-linear selection function, $\phi(\cdot)$, the performance degradation of LowRAMP is mainly due to the fact the second-order Taylor series approximation of the output channel $[R]$, around $z_{ij} = 0$, is not accurate high SNR values (which is a crucial condition to solve the matrix completion problem).

Fig. 13 shows the phase transition diagrams in the NRMSE for both low- and high-SNR regimes at a fixed selection rate of 20%. In the large-rank limit, we observe a low detectability regime when the SNR is in $[0, 10]$ dB. This result is consistent with the non-negligible uncertainty of estimators [31] to conduct inference in the low-SNR regime for noisy matrix completion.

VI. CONCLUSION

In this work, we introduced a new algorithm, dubbed BiG-VAMP, to solve the generalized bilinear recovery problem based on the approximate message passing paradigm while treating both the MMSE and MAP inference problems in a unified framework. We described how BiG-VAMP provides a broader solution to the bilinear recovery problem under different structured matrices beyond the “low rank” structure heavily investigated in the existing literature. In particular, our numerical results for applications in matrix-factorization, dictionary learning, and matrix completion demonstrated that BiG-VAMP exhibits the best reconstruction performance under non-Gaussian priors as compared to existing state-of-the-art algorithms such as BiG-AMP, BAd-VAMP, and LowRAMP. Additionally, we derived the state evolution equations for BiG-VAMP and characterized its phase transition for the matrix completion problem.

ACKNOWLEDGMENT

This work was supported by the Discovery Grants (DG) program from the Natural Science and Engineering Council of Canada (NSERC).

REFERENCES

[1] R. Gribonval, R. Jenatton, and F. Bach, “Sparse and spurious: dictionary learning with noise and outliers,” IEEE Transactions on Information Theory, vol. 61, no. 11, pp. 6298–6319, 2015.
[2] A. Montanari, Y. Eldar, and G. Kutyniok, “Graphical models concepts in compressed sensing,” Compressed Sensing: Theory and Applications, pp. 394–438, 2012.
[3] H. Zou, T. Hastie, and R. Tibshirani, “Sparse principal component analysis,” Journal of computational and graphical statistics, vol. 15, no. 2, pp. 265–286, 2006.
[4] Y. Koren, R. Bell, and C. Volinsky, “Matrix factorization techniques for recommender systems,” Computer, no. 8, pp. 30–37, 2009.
[5] R. Matsushita and T. Tanaka, “Low-rank matrix reconstruction and clustering via approximate message passing,” in Advances in Neural Information Processing Systems, 2013, pp. 917–925.
[6] L. Wang, Z. Zhang, and D. Dunson, “Symmetric bilinear regression for signal subgraph estimation,” IEEE Transactions on Signal Processing, vol. 67, no. 7, pp. 1929–1940, 2019.
[7] Z. Lin, M. Chen, and Y. Ma, “The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices,” arXiv preprint arXiv:1009.5055, 2010.
[8] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein et al., “Distributed optimization and statistical learning via the alternating direction method of multipliers,” Foundations and Trends® in Machine learning, vol. 3, no. 1, pp. 1–122, 2011.
[9] S. D. Babacan, M. Luessi, R. Molina, and A. K. Katsaggelos, “Sparse bayesian methods for low-rank matrix estimation,” *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 3964–3977, 2012.

[10] D. L. Donoho, A. Maleki, and A. Montanari, “Message-passing algorithms for compressed sensing,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18914–18919, 2009.

[11] Y. C. Eldar and G. Kutyniok, *Compressed sensing: theory and applications*. Cambridge university press, 2012.

[12] S. Rangan, “Generalized approximate message passing for estimation with random linear mixing,” in *2011 IEEE International Symposium on Information Theory Proceedings*, 2011, pp. 2168–2172.

[13] M. Bayati and A. Montanari, “The dynamics of message passing on dense graphs, with applications to compressed sensing,” *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 764–785, 2011.

[14] J. Barbier, F. Krzakala, N. Macris, L. Miolane, and L. Zdeborová, “Optimal errors and phase transitions in high-dimensional generalized linear models,” *Proceedings of the National Academy of Sciences*, vol. 116, no. 12, pp. 5451–5460, 2019.

[15] M. Mezard and A. Montanari, *Information, physics, and computation*. Oxford University Press, 2009.

[16] S. Rangan, P. Schniter, and A. K. Fletcher, “Vector approximate message passing,” in *2017 IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 1588–1592.

[17] P. Schniter, S. Rangan, and A. K. Fletcher, “Vector approximate message passing for the generalized linear model,” in *2016 50th Asilomar Conference on Signals, Systems and Computers*, IEEE, 2016, pp. 1525–1529.

[18] S. Rangan, P. Schniter, and A. K. Fletcher, “Vector approximate message passing,” *IEEE Transactions on Information Theory*, vol. 65, no. 10, pp. 6664–6684, 2019.

[19] J. Ma and L. Ping, “Orthogonal amp,” *IEEE Access*, vol. 5, pp. 2020–2033, 2017.

[20] T. Lesieur, F. Krzakala, and L. Zdeborová, “Phase transitions in sparse pca,” in *2015 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2015, pp. 1635–1639.

[21] J. T. Parker, P. Schniter, and V. Cevher, “Bilinear generalized approximate message passing—part i: Derivation,” *IEEE Transactions on Signal Processing*, vol. 62, no. 22, pp. 5839–5853, 2014.

[22] J. T. Parker and P. Schniter, “Parametric bilinear generalized approximate message passing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 4, pp. 795–808, 2016.

[23] S. Sarkar, A. K. Fletcher, S. Rangan, and P. Schniter, “Bilinear recovery using adaptive vector-amp,” *IEEE Transactions on Signal Processing*, vol. 67, no. 13, pp. 3383–3396, 2019.

[24] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[25] A. M. Tulino, S. Verdú et al., “Random matrix theory and wireless communications,” *Foundations and Trends® in Communications and Information Theory*, vol. 1, no. 1, pp. 1–182, 2004.

[26] P. Schniter, “BiGAMP.” [Online]. Available: https://sourceforge.net/projects/gampmatlab/.

[27] S. Sarkar, “Bilinear Adaptive Vector Approximate Message Passing,” Jul. 2019. [Online]. Available: https://github.com/sbrsarkar/BAdVAMP.

[28] A. Perry, A. Wein, A. Bandeira, and A. Moitra, “Message-Passing Algorithms for Synchronization Problems over Compact Group,” *Communications on Pure and Applied Mathematics*, 10 2016.

[29] A. Mezghani and A. L. Swindlehurst, “Blind Estimation of Sparse Broadband Massive MIMO Channels With Ideal and One-bit ADCs,” *IEEE Transactions on Signal Processing*, vol. 66, no. 11, pp. 2972–2983, 2018.

[30] F. Krzakala, “Low rank Matrix Factorization with AMP,” Aug. 2015. [Online]. Available: https://github.com/krzakala/LowRAMP.

[31] Y. Chen, J. Fan, C. Ma, and Y. Yan, “Inference and uncertainty quantification for noisy matrix completion,” *Proceedings of the National Academy of Sciences*, vol. 116, no. 46, pp. 22931–22937, 2019.