Inference for Generalized Linear Models via Alternating Directions and Bethe Free Energy Minimization

Sundeep Rangan, *Fellow, IEEE*, Alyson K. Fletcher, *Member, IEEE*, Philip Schniter, *Fellow, IEEE*, and Ulugbek S. Kamilov, *Member, IEEE*

***Abstract*— Generalized linear models, where a random vector x is observed through a noisy, possibly nonlinear, function of a linear transform z = Ax, arise in a range of applications in nonlinear filtering and regression. Approximate message passing (AMP) methods, based on loopy belief propagation, are a promising class of approaches for approximate inference in these models. AMP methods are computationally simple, general, and admit precise analyses with testable conditions for optimality for large i.i.d. transforms A. However, the algorithms can diverge for general A. This paper presents a convergent approach to the generalized AMP (GAMP) algorithm based on direct minimization of a large-system limit approximation of the Bethe free energy (LSL-BFE). The proposed method uses a double-loop procedure, where the outer loop successively linearizes the LSL-BFE and the inner loop minimizes the linearized LSL-BFE using the alternating direction method of multipliers (ADMM). The proposed method, called ADMM-GAMP, is similar in structure to the original GAMP method, but with an additional least-squares minimization. It is shown that for strictly convex, smooth penalties, ADMM-GAMP is guaranteed to converge to a local minimum of the LSL-BFE, thus providing a convergent alternative to GAMP that is stable under arbitrary transforms. Simulations are also presented that demonstrate the robustness of the method for non-convex penalties as well.**

***Index Terms*— Belief propagation, ADMM, message passing, variational optimization, generalized linear models.**

Manuscript received January 8, 2015; revised May 2, 2016; accepted September 19, 2016. Date of publication October 19, 2016; date of current version December 20, 2016. S. Rangan was supported by the National Science Foundation under Grant 1116589, Grant 1302336, Grant 1564142, and Grant 1547332. A. K. Fletcher was supported in part by the National Science Foundation under Grant 1254204 and in part by the Office of Naval Research under Grant N00014-15-1-2677. P. Schniter was supported by the National Science Foundation under Grant CCF-1218754, Grant CCF-1018368, and Grant CCF-1527162. U. S. Kamilov was supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC under Grant 267439. This paper was presented at the 2015 IEEE Symposium on Information Theory.

S. Rangan is with the Department of Electrical and Computer Engineering, New York University, Brooklyn, NY 11201, USA (e-mail: srangan@nyu.edu).

A. K. Fletcher is with the Departments of Statistics, Mathematics, and Electrical Engineering, University of California at Los Angeles, Los Angeles, CA 90095, USA (e-mail: akfletcher@ucla.edu).

P. Schniter is with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH, USA (e-mail: schniter.1@osu.edu).

U. S. Kamilov was with the Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne, Lausanne 1015, Switzerland. He is now is with Mitsubishi Electric Research Laboratories, Cambridge, MA 02139 USA (e-mail: kamilov@merl.com).

Communicated by A. Montanari, Associate Editor for Statistical Learning.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2016.2619373

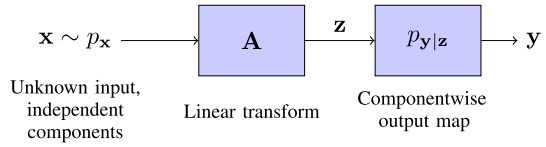

Fig. 1. Generalized Linear Model (GLM) where an unknown random vector $\mathbf{x}$ is observed via a linear transform followed by componentwise likelihood to yield a measurement vector $\mathbf{y}$.

I. Introduction

Consider the problem of estimating a random vector $\mathbf{x} \in \mathbb{R}^n$ from observations $\mathbf{y} \in \mathbb{R}^m$ as shown in Fig. 1. The unknown vector is assumed to have a prior density of the form $p(\mathbf{x}) = \mathrm{e}^{-f_x(\mathbf{x})}$ and the observations $\mathbf{y} \in \mathbb{R}^m$ are described by a likelihood function of the form $p(\mathbf{y}|\mathbf{x}) = \mathrm{e}^{-f_z(\mathbf{A}\mathbf{x},\mathbf{y})}$ for some known transform $\mathbf{A} \in \mathbb{R}^{m\times n}$. In statistics, this model is a special case of a generalized linear model (GLM) [1], [2] and arises in a range of applications including statistical regression, filtering, inverse problems, and nonlinear forms of compressed sensing. The posterior density of $\mathbf{x}$ given $\mathbf{y}$ in the GLM model is given by

$$p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}) = \frac{1}{Z(\mathbf{y})} \exp\big[-f_x(\mathbf{x}) - f_z(\mathbf{A}\mathbf{x}, \mathbf{y})\big], \tag{1}$$

where $Z(\mathbf{y})$ is a normalization constant. In the sequel, we will often omit the dependence on $\mathbf{y}$ and simply write

$$p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}) = \frac{1}{Z} \exp\big[-f_x(\mathbf{x}) - f_z(\mathbf{A}\mathbf{x})\big], \tag{2}$$

so that the dependence on $\mathbf{y}$ in the function $f_z(\cdot)$ and the normalization constant Z is implicit. In this work, we consider the inference problem of estimating the posterior marginal distributions, $p_{x_j|\mathbf{y}}(x_j|\mathbf{y})$. From these posterior marginals, one can compute the posterior means and variances

$$\widehat{x}_j \triangleq \mathbb{E}(x_j|\mathbf{y}), \tag{3a}$$

$$\tau_{x_j} \triangleq \mathrm{var}(x_j|\mathbf{y}). \tag{3b}$$

We study this inference problem in the case where the functions f_x and f_z are separable, in that they are of the form

$$f_x(\mathbf{x}) = \sum_{j=1}^{n} f_{x_j}(x_j), \tag{4a}$$

$$f_z(\mathbf{z}) = \sum_{i=1}^{m} f_{z_i}(z_i), \tag{4b}$$

0018-9448 © 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

for some scalar functions f_{x_j} and f_{z_i}. The separability assumption (4a) corresponds to the components in $\mathbf{x}$ being *a priori* independent. Recalling the implicit dependence of f_z on $\mathbf{y}$, the separability assumption (4b) corresponds to the observations $\mathbf{y}$ being conditionally independent given the transform outputs $\mathbf{z} \triangleq \mathbf{A}\mathbf{x}$.

For posterior densities of the form (2), there are several computationally efficient methods to find the *maximum a posteriori* (MAP) estimate, which is given by

$$\begin{aligned}\widehat{\mathbf{x}} &= \arg\max_{\mathbf{x}} p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y}) \\ &= \arg\min_{\mathbf{x}} \big[f_x(\mathbf{x}) + f_z(\mathbf{A}\mathbf{x}) \big]. \end{aligned} \tag{5}$$

Under the separability assumptions (4), the MAP minimization (5) admits a factorizable dual decomposition that can be exploited by a variety of approaches, including variants of the iterative shrinkage and thresholding algorithm (ISTA) [3]–[8] and the alternating direction method of multipliers (ADMM) [9]–[12].

In contrast, the inference problem of estimating the posterior marginals $p(x_j|\mathbf{y})$ and the corresponding minimum mean squared error (MMSE) estimates (3a) is often more difficult—even in the case when f_x and f_z are convex. As a simple example, consider the case where $f_x(\mathbf{x}) = 0$ and each $f_{z_i}(z_i)$ constrains z_i to belong to some interval, so that $f_z(\mathbf{A}\mathbf{x})$ constrains $\mathbf{x}$ to belong to some polytope. The MAP estimate (5) is then given by any point in the polytope. Such a point can can be computed via a linear program. However, the MMSE estimate (3a) is the centroid of the polytope which is, in general, #P-hard to compute [13].

GLM inference methods often use a penalized quasi-likelihood method [14] or some form of Gibbs sampling [15], [16]. In recent years, Bayesian forms of approximate message passing (AMP) have been considered as a potential alternate class of methods for approximate inference in GLMs [17]–[21]. AMP methods are based on Gaussian and quadratic approximations to loopy belief propagation (loopy BP) in graphical models and are both computationally simple and applicable to arbitrary separable penalty functions f_x and f_z. In addition, for certain large i.i.d. transforms $\mathbf{A}$, they have the benefit that the behavior of the algorithm can be exactly predicted by a state evolution analysis, which then provides testable conditions for Bayes optimality [21]–[23].

Unfortunately, for general $\mathbf{A}$, AMP methods may diverge [24], [25]—a situation that is not surprising given that AMP is based on loopy BP, which also may diverge. Several recent modifications have been proposed to improve the stability of AMP, including damping [24], sequential updating [26], and adaptive damping [27]. However, while these methods appear to perform well empirically, little has been proven rigorously about their convergence.

The main goal of this paper is to provide a provably convergent approach to AMP. We focus on the generalized AMP (GAMP) method of [21], which allows arbitrary separable functions for both f_x and f_z. Our approach to stabilizing GAMP is based on reconsidering the inference problem as a type of free-energy minimization. Specifically, it is known that GAMP can be considered as an iterative procedure for minimizing a large-system-limit approximation of the so-called Bethe Free Energy (BFE) [28], [29], which we abbreviate as "LSL-BFE" in the sequel. The BFE also plays a central role in loopy BP [30], and we review both the BFE and LSL-BFE in Section III.

In contrast to GAMP, which *implicitly* minimizes the LSL-BFE through an approximation of the sum-product algorithm, our proposed approach *explicitly* minimizes the LSL-BFE. We propose a double-loop algorithm, similar to the well-known Convex Concave Procedure (CCCP) [31]. Specifically, the outer loop of our method successively approximates the LSL-BFE by partially linearizing the LSL-BFE around the current belief estimate, while the inner loop minimizes the linearized LSL-BFE using ADMM [9]. Similar applications of ADMM have also been proposed for related free-energy minimizations [32]. Interestingly, our proposed double-loop algorithm, which we dub ADMM-GAMP, is similar in structure to the original GAMP method of [21], but with an additional least squares optimization. We discuss these differences in detail in Section VIII.

Our main theoretical result shows that, for strictly convex penalties, the proposed ADMM-GAMP algorithm is guaranteed to converge to at least a local minimum of the LSL-BFE. In this way, we obtain a variant of the GAMP method with a provable convergence guarantee for arbitrary transforms $\mathbf{A}$. In addition, using hardening arguments similar to [33] and [34], we show that our ADMM-GAMP can also be applied to the MAP estimation problem, in which case we can obtain global convergence for strictly convex, smooth penalties. Also, while our theory requires convex penalties, we present simulations that show robust behavior even in non-convex cases.

II. The Generalized Linear Model

Before describing our optimization approach, it is useful to briefly provide some examples of the model (1) to illustrate the generality of the framework. As a first simple example, consider a simple linear model

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\epsilon}, \tag{6}$$

where $\mathbf{A}$ is a known matrix, $\mathbf{x}$ is an unknown vector and $\boldsymbol{\epsilon}$ is a noise vector. In statistics, $\mathbf{A}$ would be the data matrix with predictors, $\mathbf{x}$ would be the vector of regression coefficients, $\mathbf{y}$ the vector of target or response variables and $\boldsymbol{\epsilon}$ would represent the model errors. To place this model in the framework of this paper, we must impose a prior $p(\mathbf{x})$ on $\mathbf{x}$ and model the noise $\boldsymbol{\epsilon}$ as a random vector independent of $\mathbf{A}$ and $\mathbf{x}$. Under these assumptions, the posterior density of $\mathbf{x}$ given $\mathbf{y}$ will be of the form (1) if we define

$$f_x(\mathbf{x}) := -\ln p(\mathbf{x}) \tag{7a}$$
$$f_z(\mathbf{z}) := -\ln p(\mathbf{y}|\mathbf{z}) = -\ln p_\epsilon(\mathbf{y} - \mathbf{z}). \tag{7b}$$

The separability assumption (4) will be valid if the components of x_j are w_i are independent so the prior and noise density factorizes as

$$p(\mathbf{x}) = \prod_{j=1}^{n} p(x_j), \quad p(\boldsymbol{\epsilon}) = \prod_{i=1}^{m} p(\epsilon_i).$$

If the output noise $\boldsymbol{\epsilon}$ is Gaussian with independent components $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$, the output factor $f_z(\mathbf{z})$ in (7) has a quadratic cost,

$$f_z(\mathbf{z}) = \frac{1}{2\sigma_\epsilon^2}\|\mathbf{y} - \mathbf{z}\|^2.$$

Similarly, if $\mathbf{x}$ has a Gaussian prior with $\mathcal{N}(0, \sigma_x^2\mathbf{I})$, the input factor $f_x(\mathbf{x})$ will be given by

$$f_x(\mathbf{x}) = \frac{1}{2\sigma_\epsilon^2}\|\mathbf{x}\|^2.$$

Note that the estimation in this quadratic case would be given by standard least squares estimation.

However, much more general models are possible. For example, for Bayesian forms of compressed sensing problems, one can impose a sparse prior $p(\mathbf{x})$ such as a Bernoulli-Gaussian or a heavy-tailed density [35].

Also, for the output, any likelihood $p(\mathbf{y}|\mathbf{z})$ that factorizes as $\prod_i p(y_i|z_i)$ can be incorporated. This model would occur, for example, under any output nonlinearities as considered in [36],

$$y_i = \phi_i(z_i) + \epsilon_i,$$

where $\phi_i(z_i)$ is a known, nonlinear function and ϵ_i is noise. The model can also include logistic regression [37] where $y_i \in \{0, 1\}$ is a binary class variable and

$$P(y_i = 1|z_i) = 1 - P(y_i = 0|z_i) = \sigma(z_i),$$

for some sigmoidal function $\sigma(z)$. One-bit and quantized compressed sensing [38] as well as Poisson output models [39] can also be easily modeled.

III. Bethe Free Energy Minimization

We next provide a brief review of the Bethe Free Energy (BFE) minimization approach to estimation of marginal densities in GLMs. A more complete treatment of this topic, along with related ideas in variational inference, can be found in [30] and [40].

For a generic density $p(\mathbf{x}|\mathbf{y})$, exact computation of the marginal densities $p(x_j|\mathbf{y})$ is difficult, because it involves a potentially high-dimensional integration. BFE minimization provides an approximate approach to marginal density computation for the case when the joint density admits a factorizable structure of the form

$$p(\mathbf{x}|\mathbf{y}) \propto \prod_{\ell=1}^{L} \psi_\ell(\mathbf{x}_{\alpha(\ell)}|\mathbf{y}), \tag{8}$$

where, for each ℓ, $\mathbf{x}_{\alpha(\ell)}$ is a sub-vector of $\mathbf{x}$ created from indices in the subset $\alpha(\ell)$ and ψ_ℓ is a potential function on that sub-vector. In this case, BFE minimization aims to compute the vectors of densities

$$\mathbf{b} \triangleq [b_1, \ldots, b_n]^{\mathsf{T}} \quad \text{and} \quad \mathbf{q} \triangleq [q_1, \ldots, q_L]^{\mathsf{T}},$$

where $b_j(x_j)$ represents an estimate of the marginal density $p(x_j|\mathbf{y})$ and where $q_\ell(\mathbf{x}_{\alpha(\ell)})$ represents an estimate of the joint density $p(\mathbf{x}_{\alpha(\ell)}|\mathbf{y})$ on the sub-vector $\mathbf{x}_{\alpha(\ell)}$. These density estimates, often called "beliefs," are computed using an optimization of the form

$$(\widehat{\mathbf{b}}, \widehat{\mathbf{q}}) = \arg\min_{(\mathbf{b},\mathbf{q})\in E} J(\mathbf{b}, \mathbf{q}), \tag{9}$$

where $J(\mathbf{b}, \mathbf{q})$ is the BFE given by

$$J(\mathbf{b}, \mathbf{q}) \triangleq \sum_{\ell=1}^{L} D(q_\ell \| \psi_\ell) + \sum_{j=1}^{n} (n_j - 1) H(b_j); \tag{10}$$

where $D(q_\ell\|\psi_\ell)$ is the KL divergence,

$$D(a\|b) \triangleq \int a(x) \ln\left[\frac{a(x)}{b(x)}\right] \mathrm{d}x; \tag{11}$$

where $H(b_j)$ is the entropy or differential entropy; and where (for each j) n_j is the number of factors ℓ such that $j \in \alpha(\ell)$. The BFE minimization (9) is performed over the set E of all $(\mathbf{b}, \mathbf{q})$ whose components satisfy a particular "matching" condition: for each $j \in \alpha(\ell)$, the marginal density of x_j within the belief $q_\ell(\mathbf{x}_{\alpha(\ell)})$ must agree with the belief $b_j(x_j)$. That is, the set E contains all $(\mathbf{b}, \mathbf{q})$ such that

$$\int q_\ell(\mathbf{x}_{\alpha(\ell)})\, \mathrm{d}\mathbf{x}_{\alpha(\ell)\setminus j} = b_j(x_j), \text{ for all } \ell, j, \tag{12}$$

where the integration is over the components in the sub-vector $\mathbf{x}_{\alpha(\ell)}$ holding x_j constant. Note that E imposes a set of linear constraints on the belief vectors $\mathbf{b}$ and $\mathbf{q}$.

The BFE minimization exactly recovers the true marginals in certain cases (e.g., when the factor graph has no cycles) and provides good estimates in many other scenarios as well; see [40] for a complete discussion. In addition, due to its separable structure, the BFE can be typically minimized "locally," by solving a set of minimizations over the densities $\mathbf{b}$ and $\mathbf{q}$. When the cardinalities of the subsets $\alpha(\ell)$ are small, these local minimizations may involve much less computation than directly calculating the marginals of the full joint density $p(\mathbf{x}|\mathbf{y})$. In fact, the classic result of [30] is that loopy belief propagation can be interpreted precisely as one type of iterative local minimization of the BFE.

For the GLM in Section I, the separability assumption (4a) allows us to write the density (2) in the factorized form (8) using the $L = n + m$ potentials

$$\psi_j(x_j) = \exp(-f_{x_j}(x_j)), \quad j = 1, \ldots, n, \tag{13a}$$

$$\psi_{n+i}(\mathbf{x}) = \exp(-f_{z_i}(\mathbf{a}_i^{\mathsf{T}}\mathbf{x})), \quad i = 1, \ldots, m, \tag{13b}$$

where $\mathbf{a}_i^{\mathsf{T}}$ is the i-th row of $\mathbf{A}$. Note that, if $\mathbf{A}$ is a non-sparse matrix, then $f_{z_i}(\mathbf{a}_i^{\mathsf{T}}\mathbf{x})$ depends on all components in the vector $\mathbf{x}$. In this case, the application of traditional loopy BP—as described for example in [41]—does not generally yield a significant computational improvement.

The GAMP algorithm from [21] can be seen as an approximate BFE minimization method for GLMs with possibly dense transforms $\mathbf{A}$. Specifically, it was shown in [28] that the stationary points of GAMP coincide with the local minima of the constrained optimization

$$(\widehat{b}_x, \widehat{b}_z) \triangleq \arg\min_{b_x, b_z} J(b_x, b_z) \text{ such that} \tag{14a}$$

$$\mathbb{E}(\mathbf{z}|b_z) = \mathbf{A}\mathbb{E}(\mathbf{x}|b_x) \tag{14b}$$

where b_x and b_z are product densities, i.e.,

$$b_x(\mathbf{x}) = \prod_{j=1}^{n} b_{x_j}(x_j), \qquad b_z(\mathbf{z}) = \prod_{i=1}^{m} b_{z_i}(z_i), \tag{15}$$

and the objective function $J(b_x, b_z)$ is given by

$$J(b_x, b_z) \triangleq D(b_x \| \mathrm{e}^{-f_x}) + D(b_z \| Z_z^{-1} \mathrm{e}^{-f_z}) + H\big(\mathrm{var}(\mathbf{x}|b_x), \mathrm{var}(\mathbf{z}|b_z)\big), \tag{16}$$

$$H(\boldsymbol{\tau}_x, \boldsymbol{\tau}_z) \triangleq \frac{1}{2}\sum_{i=1}^{m}\left[\frac{\tau_{z_i}}{\sum_{j=1}^{n} S_{ij}\tau_{x_j}} + \ln\left(2\pi \sum_{j=1}^{n} S_{ij}\tau_{x_j}\right)\right], \tag{17}$$

$$\boldsymbol{\tau}_x \triangleq (\tau_{x_1}, \ldots, \tau_{x_n})^{\mathsf{T}}, \quad \tau_{x_j} \triangleq \mathrm{var}(x_j|b_{x_j}), \tag{18}$$

$$\boldsymbol{\tau}_z \triangleq (\tau_{z_1}, \ldots, \tau_{z_m})^{\mathsf{T}}, \quad \tau_{z_i} \triangleq \mathrm{var}(z_i|b_{z_i}), \tag{19}$$

$$S_{ij} = [\mathbf{S}]_{ij} \triangleq [\mathbf{A}]_{ij}^2 \quad \forall i, j. \tag{20}$$

Above, and in the sequel, we use $\mathbb{E}(\mathbf{x}|b_x) \in \mathbb{R}^n$ to denote the expectation of $\mathbf{x}$ under $\mathbf{x} \sim b_x$, and we use $\mathrm{var}(\mathbf{x}|b_x) \in \mathbb{R}_+^n$ to denote the vector whose jth entry is the variance of x_j under $\mathbf{x} \sim b_x$. Note that $\mathrm{var}(\mathbf{x}|b_x)$ is not a full covariance matrix. Also, $Z_z \triangleq \int_{\mathbb{R}^m} \mathrm{e}^{-f_z(\mathbf{z})}\,\mathrm{d}\mathbf{z}$ is the scale factor that makes $Z_z^{-1}\mathrm{e}^{-f_z(\mathbf{z})}$ a valid density over $\mathbf{z} \in \mathbb{R}^m$. Although it is not essential for this paper, we note that $H\big(\mathrm{var}(\mathbf{x}|b_x), \mathrm{var}(\mathbf{z}|b_z)\big)$ is an upper bound on the differential entropy of b_z that is tight when b_z has independent Gaussian entries with variances $\boldsymbol{\tau}_z = \mathbf{S}\boldsymbol{\tau}_x$. It was then shown in [29] that the objective function in (16) can be interpreted as an approximation of the BFE for the GLM from Section I in a certain large-system limit, where $m, n \to \infty$ and $\mathbf{A}$ has i.i.d. entries. We thus call the approximate BFE in (16) the *large-system limit Bethe Free Energy* or LSL-BFE.

Similar to the case of loopy BP, it has been shown in [28] and [29] that the stationary points of (14) are precisely the fixed points of sum-product GAMP. Thus, GAMP can be interpreted as an iterative procedure to find local minima of the LSL-BFE, much in the same way that loopy BP can be interpreted as an iterative way to find local minima of the BFE. The trouble with GAMP, however, is that it does not always converge (see, e.g., the negative results in [24], [25], and [27]). The situation is similar to the case of loopy BP. Although several modifications of GAMP have been proposed with the goal of improving convergence, such as damping [24], sequential updating [26], and adaptive damping [27], a globally convergent GAMP modification remains elusive.

IV. Minimization via Iterative Linearization

Our approach to finding a convergent algorithm for minimizing the constrained LSL-BFE employs a generalization of the convex-concave procedure (CCCP) of [31] that we will refer to as *Minimization via Iterative Linearization*.

A. The Convex-Concave Procedure

We first briefly review the CCCP. Observe that, in the BFE (10), the $D(q_\ell \| \psi_\ell)$ terms are convex in q_ℓ and the $H(b_j)$ terms are concave in b_j. Thus, the BFE (10) can be written as a sum of terms

$$J(\mathbf{b}, \mathbf{q}) = f(\mathbf{q}) + h(\mathbf{b}),$$

where f is convex and h is concave. The CCCP finds a sequence of estimates of a BFE minimizer $(\widehat{\mathbf{b}}, \widehat{\mathbf{q}})$ by iteratively linearizing the concave part of this function, i.e.,

$$(\mathbf{b}^k, \mathbf{q}^k) = \operatorname*{arg\,min}_{(\mathbf{b},\mathbf{q})\in E} f(\mathbf{b}) + (\boldsymbol{\gamma}^k)^{\mathsf{T}}\mathbf{q}, \tag{21a}$$

$$\boldsymbol{\gamma}^{k+1} = \frac{\partial h(\mathbf{q}^k)}{\partial \mathbf{q}}, \tag{21b}$$

where $\partial h(\mathbf{q}^k)/\partial \mathbf{q}$ denotes the gradient of h at $\mathbf{q}^k$. The resulting procedure is often called a "double-loop" algorithm, since each iteration involves a minimization (21a) that is itself usually performed by an iterative procedure. Because f is convex and the constraint $(\mathbf{b}, \mathbf{q}) \in E$ is linear, the minimization problem (21a) is convex. Thus, the CCCP converts the non-convex BFE minimization to a sequence of convex minimizations. In fact, it can be shown that the CCCP will monotonically decrease the BFE for arbitrary convex f and concave h [31].

B. Minimization via Iterative Linearization

For the LSL-BFE, it is not convenient to decompose the objective function into a convex term plus a concave term. To handle problems like LSL-BFE minimization, we consider optimization problems of the form

$$\min_{\mathbf{b}\in B} J(\mathbf{b}), \quad J(\mathbf{b}) = f(\mathbf{b}) + h(\mathbf{g}(\mathbf{b})), \tag{22}$$

where now $\mathbf{b}$ is a vector in a Hilbert space $\mathcal{H}_b$, B is a closed affine subspace of $\mathcal{H}_b$, $f : \mathcal{H}_b \to \mathbb{R}$ is a convex functional, $\mathbf{g} : \mathcal{H}_b \to \mathbb{R}^p$ is a mapping from $\mathcal{H}_b$ to $\mathbb{R}^p$ for some $p \in \mathbb{N}$, and $h : \mathbb{R}^p \to \mathbb{R}$ is an arbitrary functional. Below, we use $\boldsymbol{\tau} \in \mathbb{R}^p$ to denote the input to h. Note that the functionals h and $h(\mathbf{g}(\cdot))$ may be neither concave nor convex.

To solve (22), we propose the iterative procedure shown in Algorithm 1, which is reminiscent of the CCCP. At each iteration k, an estimate $\mathbf{b}^k$ of $\arg\min_{\mathbf{b}\in B} J(\mathbf{b})$ is computed by minimizing the functional

$$J(\mathbf{b}, \boldsymbol{\gamma}^k) \triangleq f(\mathbf{b}) + (\boldsymbol{\gamma}^k)^{\mathsf{T}}\mathbf{g}(\mathbf{b}), \tag{23}$$

where $\boldsymbol{\gamma}^k \in \mathbb{R}^p$ is a "damped" version of the gradient $\partial h(\mathbf{g}(\mathbf{b}))/\partial \mathbf{b}$. In particular, when the *damping parameter* θ^k is set to unity, the linearization vector is exactly equal to the gradient at $\mathbf{b}^k$, i.e., $\boldsymbol{\gamma}^k = \partial[h(\mathbf{g}(\mathbf{b}^k))]/\partial \mathbf{b}$, similar to CCCP. However, in Algorithm 1, we have the option of setting $\theta^k < 1$, which has the effect of slowing the update on $\boldsymbol{\gamma}^k$. We will see that, by setting $\theta^k < 1$, we can guarantee convergence when h and/or $h(\mathbf{g}(\cdot))$ is non-concave.

C. Convergence of Minimization via Iterative Linearization

Observe that when f is convex, h is concave, $\mathcal{H}_b = \mathbb{R}^p$ (as when x_j are discrete variables), $\mathbf{g}$ is the identity map (i.e., $\mathbf{g}(\mathbf{b}) = \mathbf{b}$), and there is no damping (i.e., $\theta^k = 1\ \forall k$), Algorithm 1 reduces to the CCCP. However, we are interested in possibly non-concave $h(\mathbf{g}(\cdot))$, in which case we cannot directly apply the results of [31]. We instead consider the following alternate conditions.

Assumption 1: Consider the optimization problem (22), and suppose that the functions f, $\mathbf{g}$, and h have components

Algorithm 1 Minimization via Iterative Linearization

Require: Optimization problem (22).

$k \leftarrow 0$
Select initial linearization $\boldsymbol{\gamma}^0$
repeat
  {Minimize the linearized function}
  $\mathbf{b}^k \leftarrow \arg\min_{\mathbf{b}\in B} J(\mathbf{b}, \boldsymbol{\gamma}^k)$

  {Update the linearization}
  Select a damping parameter $\theta^k \in (0, 1]$
  $\boldsymbol{\tau}^k \leftarrow \mathbf{g}(\mathbf{b}^k)$
  $\boldsymbol{\gamma}^{k+1} \leftarrow (1-\theta^k)\boldsymbol{\gamma}^k + \theta^k \dfrac{\partial h(\boldsymbol{\tau}^k)}{\partial \boldsymbol{\tau}}$
until Terminated

that are twice differentiable with uniformly bounded second derivatives. Also, assume that there exists a convex set Γ such that, for all $\boldsymbol{\gamma} \in \Gamma$:

1) The minimization of the linearized function,
$$\widehat{\mathbf{b}}(\boldsymbol{\gamma}) \triangleq \arg\min_{\mathbf{b}\in B} J(\mathbf{b}, \boldsymbol{\gamma}) \tag{24}$$
exists and is unique.
2) At each minimum, the linearized objective is uniformly strictly convex in the linear space B in that there exists constants c_1, c_2 with $c_2 > c_1 > 0$ such that
$$c_1\|\mathbf{u}\|^2 \le \mathbf{u}^{\mathsf{T}}\mathbf{H}(\boldsymbol{\gamma})\mathbf{u} \le c_2\|\mathbf{u}\|^2, \ \forall \mathbf{u} : \widehat{\mathbf{b}}(\boldsymbol{\gamma}) + \mathbf{u} \in B, \tag{25}$$
where $\mathbf{H}(\boldsymbol{\gamma})$ is the Hessian of J with respect to $\mathbf{b}$ at $\widehat{\mathbf{b}}(\boldsymbol{\gamma})$, i.e.,
$$\mathbf{H}(\boldsymbol{\gamma}) \triangleq \left.\frac{\partial^2 J(\mathbf{b}, \boldsymbol{\gamma})}{\partial \mathbf{b}\, \partial \mathbf{b}^{\mathsf{T}}}\right|_{\mathbf{b}=\widehat{\mathbf{b}}(\boldsymbol{\gamma})}, \tag{26}$$
and where the constants c_1 and c_2 do not depend on $\boldsymbol{\gamma}$.
3) The gradient obeys $\partial h(\mathbf{g}(\mathbf{b}))/\partial \boldsymbol{\tau} \in \Gamma$ when $\mathbf{b} = \widehat{\mathbf{b}}(\boldsymbol{\gamma})$.

Theorem 1: Consider Algorithm 1 under Assumption 1. There exists a $\overline{\theta} \in (0, 1)$ such that if the damping parameters are selected with $0 < \theta^k \le \overline{\theta}$ for all k, and if the initialization obeys $\boldsymbol{\gamma}^0 \in \Gamma$, then $\boldsymbol{\gamma}^k \in \Gamma$ for all k and the objective monotonically decreases, i.e.,
$$J(\mathbf{b}^{k+1}) \le J(\mathbf{b}^k) \ \forall k. \tag{27}$$

Proof: See Appendix A. ∎

The most simple case where Assumption 1 holds is the setting where $f(\mathbf{b})$ is strictly convex and smooth, $g(\mathbf{b})$ is linear and $h(\boldsymbol{\tau})$ is smooth (but neither necessarily convex nor concave). Under these assumptions, $J(\mathbf{b}, \boldsymbol{\gamma})$ would be strictly convex for all $\boldsymbol{\gamma}$, thereby satisfying Assumptions (a) and (b). The assumption would also be valid for strictly convex $f(\mathbf{b})$ and convex $g(\mathbf{b})$, provided we restrict to positive $\boldsymbol{\gamma}$. In this case, to satisfy assumption (c), we would require that $\partial h(\mathbf{g}(\mathbf{b}))/\partial \boldsymbol{\tau} \ge 0$, i.e. $h(\boldsymbol{\tau})$ is increasing in each of its component. Interestingly, in the setting we will use below, $f(\mathbf{b})$ will be convex, but $g(\mathbf{b})$ will be concave. Nevertheless, we will show that the assumption will be satisfied.

D. Application to LSL-BFE Minimization

To apply Algorithm 1 to the LSL-BFE minimization (14), we first take B to be the vector of separable density pairs $\mathbf{b} = (b_x; b_z)$ satisfying the moment matching constraint
$$B = \{(b_x; b_z) \mid \mathbb{E}(\mathbf{z}|b_z) = \mathbf{A}\mathbb{E}(\mathbf{x}|b_x)\}. \tag{28}$$
Then, if we define the functions
$$f(\mathbf{b}) \triangleq f(b_x, b_z) \tag{29a}$$
$$\triangleq D(b_x\|\mathrm{e}^{-f_x}) + D(b_z\|Z_z^{-1}\mathrm{e}^{-f_z}) \tag{29b}$$
$$\mathbf{g}(\mathbf{b})^{\mathsf{T}} \triangleq [\mathrm{var}(\mathbf{x}|b_x); \mathrm{var}(\mathbf{z}|b_z)] \triangleq [\boldsymbol{\tau}_x; \boldsymbol{\tau}_z] \tag{29c}$$
$$h([\boldsymbol{\tau}_x; \boldsymbol{\tau}_z]) \triangleq H(\boldsymbol{\tau}_x, \boldsymbol{\tau}_z), \tag{29d}$$
we see that $J(b_x, b_z)$ from (16) can be cast into the form in (22). Observe that, while f is convex, the function $h(\mathbf{g}(\cdot))$ is, in general, neither convex nor concave. Thus, while the CCCP does not apply, we can apply the iterative linearization method from Algorithm 1.

We will partition the linearization vector $\boldsymbol{\gamma}$ conformally with function $\mathbf{g}$ in (29c) as
$$\boldsymbol{\gamma} = [\mathbf{1}./(2\boldsymbol{\tau}_r); \mathbf{1}./(2\boldsymbol{\tau}_p)], \tag{30}$$
where we use "./" to denote componentwise division of two vectors and ";" to denote vertical concatenation. The notation in (30) will help to clarify the connections to the original GAMP algorithm. Using the above notation, the linearized objective (23) can be written as
$$\begin{aligned} J(b_x, b_z, \boldsymbol{\tau}_r, \boldsymbol{\tau}_p) \triangleq\ & D(b_x\|\mathrm{e}^{-f_x}) + D(b_z\|Z_z^{-1}\mathrm{e}^{-f_z}) \\ & + (\mathbf{1}./(2\boldsymbol{\tau}_r))^{\mathsf{T}}\mathrm{var}(\mathbf{x}|b_x) \\ & + \big(\mathbf{1}./(2\boldsymbol{\tau}_p)\big)^{\mathsf{T}}\mathrm{var}(\mathbf{z}|b_z). \end{aligned} \tag{31}$$

Finally, we compute the gradient $h' = \frac{\partial h}{\partial \boldsymbol{\tau}}$ of the function h from (29d). Similar to $\boldsymbol{\gamma}$, we will partition the gradient into two terms,
$$\mathbf{1}./(2\overline{\boldsymbol{\tau}}_r) \triangleq \frac{\partial H(\boldsymbol{\tau}_x, \boldsymbol{\tau}_z)}{\partial \boldsymbol{\tau}_x}, \quad \mathbf{1}./(2\overline{\boldsymbol{\tau}}_p) \triangleq \frac{\partial H(\boldsymbol{\tau}_x, \boldsymbol{\tau}_z)}{\partial \boldsymbol{\tau}_z}. \tag{32}$$
From (17), the derivative of H with respect to τ_{z_i} is
$$\frac{1}{2\overline{\tau}_{p_i}} = \frac{\partial H(\boldsymbol{\tau}_x, \boldsymbol{\tau}_z)}{\partial \tau_{z_i}} = \frac{1}{2\sum_{j=1}^n S_{ij}\tau_{x_j}}. \tag{33}$$
Similarly, using the chain rule and (33), we find
$$\begin{aligned} \frac{1}{2\overline{\tau}_{r_j}} &= \frac{\partial H(\boldsymbol{\tau}_x, \boldsymbol{\tau}_z)}{\partial \tau_{x_j}} \qquad (34)\\ &= \sum_i \frac{\partial H(\boldsymbol{\tau}_x, \boldsymbol{\tau}_z)}{\partial(\sum_k S_{ik}\tau_{x_k})}\frac{\partial(\sum_k S_{ik}\tau_{x_k})}{\partial \tau_{x_j}} \\ &= \sum_{i=1}^m \frac{1}{2}\Big(-\frac{\tau_{z_i}}{\overline{\tau}_{p_i}^2} + \frac{2\pi}{2\pi\,\overline{\tau}_{p_i}}\Big)S_{ij} \\ &= \frac{1}{2}\sum_{i=1}^m S_{ij}\Big(1 - \frac{\tau_{z_i}}{\overline{\tau}_{p_i}}\Big)\frac{1}{\overline{\tau}_{p_i}}. \qquad (35) \end{aligned}$$
We can then write (33) and (35) in vector form as
$$\overline{\boldsymbol{\tau}}_p = \mathbf{S}\boldsymbol{\tau}_x, \quad \mathbf{1}./\overline{\boldsymbol{\tau}}_r = \mathbf{S}^{\mathsf{T}}\big[\big(\mathbf{1} - \boldsymbol{\tau}_z./\overline{\boldsymbol{\tau}}_p\big)./\overline{\boldsymbol{\tau}}_p\big]. \tag{36}$$

Algorithm 2 Minimizing LSL-BFE via Iterative Linearization

Require: LSL-BFE objective function (16) with a matrix $\mathbf{A}$.

1: $k \leftarrow 0$
2: Select initial linearization $\boldsymbol{\tau}_p^0$, $\boldsymbol{\tau}_r^0$.
3: **repeat**
4: {Minimize the linearized LSL-BFE}
5: $(b_x^k, b_z^k) \leftarrow \arg\min_{(b_x,b_z)\in B} J(b_x, b_z, \boldsymbol{\tau}_r^k, \boldsymbol{\tau}_p^k)$
6:
7: {Compute the gradient terms}
8: $\boldsymbol{\tau}_x^k \leftarrow \mathrm{var}(\mathbf{x}|b_x^k)$, $\boldsymbol{\tau}_z^k \leftarrow \mathrm{var}(\mathbf{z}|b_z^k)$
9: $\overline{\boldsymbol{\tau}}_p^k \leftarrow \mathbf{S}\boldsymbol{\tau}_x^k$
10: $\boldsymbol{\tau}_s^k \leftarrow (\mathbf{1} - \boldsymbol{\tau}_z^k./\overline{\boldsymbol{\tau}}_p^k)./\overline{\boldsymbol{\tau}}_p^k$
11: $\overline{\boldsymbol{\tau}}_r^k \leftarrow \mathbf{1}./(\mathbf{S}^{\mathsf{T}}\boldsymbol{\tau}_s^k)$
12:
13: {Update the linearization}
14: Select a damping parameter $\theta^k \in (0, 1]$
15: $\mathbf{1}./\boldsymbol{\tau}_r^{k+1} \leftarrow \theta^k \mathbf{1}./\overline{\boldsymbol{\tau}}_r^k + (1-\theta^k)\mathbf{1}./\boldsymbol{\tau}_r^k$
16: $\mathbf{1}./\boldsymbol{\tau}_p^{k+1} \leftarrow \theta^k \mathbf{1}./\overline{\boldsymbol{\tau}}_p^k + (1-\theta^k)\mathbf{1}./\boldsymbol{\tau}_p^k$
17: **until** Terminated

Substituting the above computations into the iterative linearization algorithm, Algorithm 1, we obtain Algorithm 2. We refer to this as the *outer loop*, since each iteration involves a minimization of the linearized LSL-BFE in line 5. We discuss this latter minimization next and show that it can itself be performed iteratively using a set of iterations that we will refer to as the *inner loop*.

We will also show shortly that, under certain convexity conditions, the conditions of Assumption 1 are satisfied, so that Algorithm 2 will converge to a local minimum of the LSL-BFE.

E. Alternative Methods

While the method proposed in this paper is based on CCCP of [31], there are other methods for direct minimization of the BFE that may apply to the LSL-BFE as well. For example, for problems with binary variables and pairwise penalty functions, [42] and [43] propose a clever re-parametrization to convert the constrained BFE minimization to an unconstrained optimization on which gradient descent can be used. Unfortunately, it is not obvious if the LSL-BFE here can admit such a re-parametrization since the penalty functions are not pairwise and the variables are not binary.

V. Inner-Loop Minimization and ADMM-GAMP

A. ADMM Principle

The outer loop algorithm, Algorithm 2, requires that in each iteration we solve a constrained optimization of the form

$$(b_x, b_z) = \underset{b_x,b_z}{\arg\min} \quad J(b_x, b_z, \boldsymbol{\tau}_r, \boldsymbol{\tau}_p) \tag{37a}$$

$$\text{s.t.} \qquad \mathbb{E}(\mathbf{z}|b_z) = \mathbf{A}\mathbb{E}(\mathbf{x}|b_x). \tag{37b}$$

We will show that this optimization can be performed by the Alternating Direction Method of Multipliers (ADMM) [9]. ADMM is a general approach to constrained optimizations of the form

$$\min_{\mathbf{w}} f(\mathbf{w}) \text{ s.t. } \mathbf{B}\mathbf{w} = \mathbf{0}, \tag{38}$$

where $f(\mathbf{w})$ is an objective function and $\mathbf{B}$ is some constraint matrix. Corresponding to this optimization, let us define the augmented Lagrangian

$$L(\mathbf{w}, \mathbf{u}; \boldsymbol{\tau}) \triangleq f(\mathbf{w}) + \mathbf{u}^{\mathsf{T}}\mathbf{B}\mathbf{w} + \frac{1}{2}\|\mathbf{B}\mathbf{w}\|_{\boldsymbol{\tau}}^2, \tag{39}$$

where $\mathbf{u}$ is a dual vector, $\boldsymbol{\tau}$ is a vector of positive weights and $\|\mathbf{x}\|_{\boldsymbol{\tau}}^2 \triangleq \sum_j x_j^2/\tau_j$. The ADMM procedure then produces a sequence of estimates for the optimization (38) through the iterations

$$\mathbf{w}^{t+1} = \arg\min_{\mathbf{w}} L(\mathbf{w}, \mathbf{u}^t; \boldsymbol{\tau}) \tag{40a}$$

$$\mathbf{u}^{t+1} = \mathbf{u}^t + \mathrm{Diag}(\mathbf{1}./\boldsymbol{\tau})\mathbf{B}\mathbf{w}^{t+1}, \tag{40b}$$

where $\mathrm{Diag}(\mathbf{d})$ creates a diagonal matrix from the vector $\mathbf{d}$. The algorithm thus alternately updates the primal variables $\mathbf{w}^t$ and dual variables $\mathbf{u}^t$. The vector $\boldsymbol{\tau}$ can be interpreted as a step-size on the primal problem and an inverse step-size on the dual problem.

The key benefit of the ADMM method is that, for any positive step-size vector $\boldsymbol{\tau}$, the procedure is guaranteed to converge to a global optimum for convex functions $f(\mathbf{w})$ under mild conditions on $\mathbf{B}$.

B. Application of ADMM to LSL-BFE Optimization

The ADMM procedure can be applied to the linearized LSL-BFE optimization (37) as follows. First, we replace the constraint $\mathbb{E}(\mathbf{z}|b_z) = \mathbf{A}\mathbb{E}(\mathbf{x}|b_x)$ with two constraints: $\mathbb{E}(\mathbf{z}|b_z) = \mathbf{A}\mathbf{v}$ and $\mathbb{E}(\mathbf{x}|b_x) = \mathbf{v}$. Variable splittings of this form are commonly used in the context of monotropic programming [44]. With this splitting, the augmented Lagrangian for the LSL-BFE (14) becomes

$$\begin{aligned} &L(b_x, b_z, \mathbf{s}, \mathbf{q}, \mathbf{v}; \boldsymbol{\tau}_p, \boldsymbol{\tau}_r) \\ &\triangleq J(b_x, b_z, \boldsymbol{\tau}_r, \boldsymbol{\tau}_p) + \mathbf{q}^{\mathsf{T}}(\mathbb{E}(\mathbf{x}|b_x) - \mathbf{v}) + \mathbf{s}^{\mathsf{T}}(\mathbb{E}(\mathbf{z}|b_z) - \mathbf{A}\mathbf{v}) \\ &\quad + \frac{1}{2}\|\mathbb{E}(\mathbf{x}|b_x) - \mathbf{v}\|_{\boldsymbol{\tau}_r}^2 + \frac{1}{2}\|\mathbb{E}(\mathbf{z}|b_z) - \mathbf{A}\mathbf{v}\|_{\boldsymbol{\tau}_p}^2, \end{aligned} \tag{41}$$

where $\mathbf{s}$ and $\mathbf{q}$ represent the dual variables. Note that the vectors $\boldsymbol{\tau}_r$ and $\boldsymbol{\tau}_p$ that appear in the linearized LSL-BFE $J(b_x, b_z, \boldsymbol{\tau}_r, \boldsymbol{\tau}_p)$ have been used for the augmentation terms (i.e., the last two terms) in (41). This choice will be critical. From (40), the resulting ADMM recursion becomes

$$(b_x^{t+1}, b_z^{t+1}) = \underset{b_x,b_z}{\arg\min} L(b_x, b_z, \mathbf{s}^t, \mathbf{q}^t, \mathbf{v}^t; \boldsymbol{\tau}_p, \boldsymbol{\tau}_r), \tag{42a}$$

$$\mathbf{s}^{t+1} = \mathbf{s}^t + \mathrm{Diag}(\mathbf{1}./\boldsymbol{\tau}_p)\big(\mathbb{E}(\mathbf{z}|b_z^{t+1}) - \mathbf{A}\mathbf{v}^{t+1}\big), \tag{42b}$$

$$\mathbf{q}^{t+1} = \mathbf{q}^t + \mathrm{Diag}(\mathbf{1}./\boldsymbol{\tau}_r)\big(\mathbb{E}(\mathbf{x}|b_x^{t+1}) - \mathbf{v}^{t+1}\big), \tag{42c}$$

$$\mathbf{v}^{t+1} = \arg\min_{\mathbf{v}} L(b_x^{t+1}, b_z^{t+1}, \mathbf{s}^{t+1}, \mathbf{q}^{t+1}, \mathbf{v}; \boldsymbol{\tau}_p, \boldsymbol{\tau}_r). \tag{42d}$$

To compute the minimization in (42a), we first note that the second and fourth terms in (41) can be rewritten as

$$\begin{aligned}
&\mathbf{q}^{\mathsf{T}}(\mathbb{E}(\mathbf{x}|b_x)-\mathbf{v})+\frac{1}{2}\|\mathbb{E}(\mathbf{x}|b_x)-\mathbf{v}\|^2_{\boldsymbol{\tau}_r}\\
&=\sum_{j=1}^{n} q_j\mathbb{E}(x_j|b_x)+\frac{\mathbb{E}^2(x_j|b_x)-2v_j\mathbb{E}(x_j|b_x)}{2\tau_{r_j}}+\text{const}\\
&\overset{(a)}{=}\sum_{j=1}^{n}\mathbb{E}\left(q_jx_j+\frac{x_j^2-2v_jx_j}{2\tau_{r_j}}\middle|b_x\right)-\frac{\tau_{x_j}}{2\tau_{r_j}}+\text{const}\\
&=\sum_{j=1}^{n}\frac{\mathbb{E}\big((x_j-[v_j-\tau_{r_j}q_j])^2\big|b_x\big)}{2\tau_{r_j}}-\frac{\tau_{x_j}}{2\tau_{r_j}}+\text{const}\\
&\overset{(b)}{=}\frac{1}{2}\mathbb{E}\big(\|\mathbf{x}-(\mathbf{v}-\boldsymbol{\tau}_r.\mathbf{q})\|^2_{\boldsymbol{\tau}_r}\big|b_x\big)-\sum_{j=1}^{n}\frac{\tau_{x_j}}{2\tau_{r_j}}+\text{const},
\end{aligned}\tag{43}$$

where in (a) we used $\mathbb{E}^2(x_j|b_x)=\mathbb{E}(x_j^2|b_x)-\text{var}(x_j|b_x)=\mathbb{E}(x_j^2|b_x)-\tau_{x_j}$; in (b) we used "." to denote componentwise multiplication between vectors; and "const" includes terms that are constant with respect to b_x and b_z. A similar development yields

$$\begin{aligned}
&\mathbf{s}^{\mathsf{T}}(\mathbb{E}(\mathbf{z}|b_z)-\mathbf{A}\mathbf{v})+\frac{1}{2}\|\mathbb{E}(\mathbf{z}|b_z)-\mathbf{A}\mathbf{v}\|^2_{\boldsymbol{\tau}_p}\\
&=\frac{1}{2}\mathbb{E}\big(\|\mathbf{z}-(\mathbf{A}\mathbf{v}-\boldsymbol{\tau}_p\mathbf{s})\|^2_{\boldsymbol{\tau}_p}\big|b_z\big)-\sum_{i=1}^{m}\frac{\tau_{z_i}}{2\tau_{p_i}}+\text{const}.
\end{aligned}\tag{44}$$

Also, note that the last two terms in (31) can be rewritten as

$$(\mathbf{1}./(2\boldsymbol{\tau}_r))^{\mathsf{T}}\,\text{var}(\mathbf{x}|b_x)=\sum_{j=1}^{n}\frac{\tau_{x_j}}{2\tau_{r_j}},\tag{45a}$$

$$\big(\mathbf{1}./(2\boldsymbol{\tau}_p)\big)^{\mathsf{T}}\,\text{var}(\mathbf{z}|b_z)=\sum_{i=1}^{m}\frac{\tau_{z_i}}{2\tau_{p_i}}.\tag{45b}$$

Substituting (31), (43), (44), and (45) into (41), and canceling terms, we get

$$\begin{aligned}
&L(b_x,b_z,\mathbf{s},\mathbf{q},\mathbf{v};\boldsymbol{\tau}_p,\boldsymbol{\tau}_r)\\
&\quad=D(b_x\|\mathrm{e}^{-f_x})+\frac{1}{2}\mathbb{E}\big(\|\mathbf{x}-(\mathbf{v}-\boldsymbol{\tau}_r.\mathbf{q})\|^2_{\boldsymbol{\tau}_r}\big|b_x\big)\\
&\qquad+D(b_z\|Z_z^{-1}\mathrm{e}^{-f_z})+\frac{1}{2}\mathbb{E}\big(\|\mathbf{z}-(\mathbf{A}\mathbf{v}-\boldsymbol{\tau}_p.\mathbf{s})\|^2_{\boldsymbol{\tau}_p}\big|b_z\big)\\
&\qquad+\text{const}\\
&\quad=\int_{\mathbb{R}^n}b_x(\mathbf{x})\ln\frac{b_x(\mathbf{x})}{\exp(-f_x(\mathbf{x})-\frac{1}{2}\|\mathbf{x}-(\mathbf{v}-\boldsymbol{\tau}_r.\mathbf{q})\|^2_{\boldsymbol{\tau}_r})}d\mathbf{x}\\
&\qquad+\int_{\mathbb{R}^m}b_z(\mathbf{z})\ln\frac{b_z(\mathbf{z})}{\exp(-f_z(\mathbf{z})-\frac{1}{2}\|\mathbf{z}-(\mathbf{A}\mathbf{v}-\boldsymbol{\tau}_p.\mathbf{s})\|^2_{\boldsymbol{\tau}_p})}d\mathbf{z}\\
&\qquad+\text{const}
\end{aligned}\tag{46}$$

$$=D(b_x\|p_x)+D(b_z\|p_z)+\text{const},\tag{47}$$

for $p_x(\mathbf{x})\propto\exp(-f_x(\mathbf{x})-\frac{1}{2}\|\mathbf{x}-(\mathbf{v}-\boldsymbol{\tau}_r.\mathbf{q})\|^2_{\boldsymbol{\tau}_r})$ and $p_z(\mathbf{z})\propto\exp(-f_z(\mathbf{z})-\frac{1}{2}\|\mathbf{z}-(\mathbf{A}\mathbf{v}-\boldsymbol{\tau}_p.\mathbf{s})\|^2_{\boldsymbol{\tau}_p})$. Therefore, the ADMM step (42a) has the solution,

$$b_x^{t+1}(\mathbf{x})\propto\exp\big(-f_x(\mathbf{x})-\tfrac{1}{2}\|\mathbf{x}-\mathbf{r}^t\|^2_{\boldsymbol{\tau}_r}\big)\tag{48a}$$

$$b_z^{t+1}(\mathbf{z})\propto\exp\big(-f_z(\mathbf{z})-\tfrac{1}{2}\|\mathbf{z}-\mathbf{p}^t\|^2_{\boldsymbol{\tau}_p}\big).\tag{48b}$$

for vectors

$$\mathbf{r}^t\triangleq\mathbf{v}^t-\boldsymbol{\tau}_r.\mathbf{q}^t\tag{49a}$$

$$\mathbf{p}^t\triangleq\mathbf{A}\mathbf{v}^t-\boldsymbol{\tau}_p.\mathbf{s}^t,\tag{49b}$$

where we use "." to denote componentwise vector multiplication. Using Bayes rule, (48a) can be interpreted as the posterior density of the random vector $\mathbf{x}$ under the prior $\mathrm{e}^{-f_x(\mathbf{x})}$ and an independent Gaussian likelihood with mean $\mathbf{r}^t$ and variance $\boldsymbol{\tau}_r$. Similarly, (48b) can be interpreted as the posterior pdf the random vector $\mathbf{z}$ under the likelihood $\mathrm{e}^{-f_z(\mathbf{z})}$ and an independent Gaussian prior with mean $\mathbf{p}^t$ and variance $\boldsymbol{\tau}_p$.

To tackle the minimization (42d), we ignore the $\mathbf{v}$-invariant components in the original augmented Lagrangian (41), after which (42d) can be reformulated as the least-squares problem

$$\begin{aligned}
\mathbf{v}^{t+1}=\arg\min_{\mathbf{v}}&\|\mathbf{z}^{t+1}+\boldsymbol{\tau}_p\mathbf{s}^{t+1}-\mathbf{A}\mathbf{v}\|^2_{\boldsymbol{\tau}_p}\\
&+\|\mathbf{x}^{t+1}+\boldsymbol{\tau}_r\mathbf{q}^{t+1}-\mathbf{v}\|^2_{\boldsymbol{\tau}_r}
\end{aligned}\tag{50}$$

using the definitions

$$\mathbf{z}^{t+1}\triangleq\mathbb{E}(\mathbf{z}|b_z^{t+1}),\quad\mathbf{x}^{t+1}\triangleq\mathbb{E}(\mathbf{x}|b_x^{t+1}).\tag{51}$$

C. The ADMM-GAMP Algorithm

Inserting the above ADMM updates into the outer loop algorithm, Algorithm 2, we obtain the so-called ADMM-GAMP method summarized in Algorithm 3. There and elsewhere, we use "." to denote componentwise vector-vector multiplication and "./" to denote componentwise vector-vector division. Note that the updates for the ADMM iteration appear under the comment "ADMM inner iteration."

Although, in principle, we should perform an infinite number of inner-loop iterations for each outer-loop iteration, Algorithm 2 is written in a more general "parallel form." In each (global) iteration t, there is one ADMM update as well as one linearization update. However, by setting the outer-loop damping parameter as $\theta^t=0$, it is possible to bypass the linearization update. Thus, we can obtain the desired double-loop behavior as follows: First, hold $\theta^t=0$ for a large number of iterations, thus running ADMM to convergence. Then, set $\theta^t>0$ for a single iteration to update the linearization. Then, hold $\theta^t=0$ for another large number of iterations, and so on. However, the parallel form of Algorithm 3 also facilitates other update schedules. For example, we could run a small number of ADMM updates for each linearization update, or we could run only one ADMM update per linearization update.

An interesting question is whether the algorithm can be run with a constant step-size $\theta^t=\theta$ for some small θ. Unfortunately, our theoretical analysis and numerical experiments consider only the double-loop implementation where several ADMM iterations are run for each outer loop update.

Another point to note in reading Algorithm 3 is that the expectation and variance operators in (31), (42b), and (42c) have been replaced by componentwise estimation functions g_x and g_z and their scaled derivatives. In particular, recall from (48) that b_x^{t+1} is fully parameterized by $(\mathbf{r}^t,\boldsymbol{\tau}_r^t)$ and that b_z^{t+1} is fully parameterized by $(\mathbf{p}^t,\boldsymbol{\tau}_p^t)$. Thus, we can write the means

Algorithm 3 ADMM-GAMP

Require: Matrix $\mathbf{A}$, estimation functions g_x and g_z.
1: $\mathbf{S} \leftarrow \mathbf{A}.\mathbf{A}$ (componentwise square)
2: Initialize $\boldsymbol{\tau}_r^0 > \mathbf{0}$, $\boldsymbol{\tau}_p^0 > \mathbf{0}$, $\mathbf{v}^0$
3: $\mathbf{q}^0 \leftarrow \mathbf{0}$, $\mathbf{s}^0 \leftarrow \mathbf{0}$
4: $t \leftarrow 0$
5: **repeat**
6: {ADMM inner iteration}
7: $\mathbf{r}^t \leftarrow \mathbf{v}^t - \boldsymbol{\tau}_r^t.\mathbf{q}^t$
8: $\mathbf{p}^t \leftarrow \mathbf{A}\mathbf{v}^t - \boldsymbol{\tau}_p^t.\mathbf{s}^t$
9: $\mathbf{x}^{t+1} \leftarrow g_x(\mathbf{r}^t, \boldsymbol{\tau}_r^t)$, $\mathbf{z}^{t+1} \leftarrow g_z(\mathbf{p}^t, \boldsymbol{\tau}_p^t)$
10: $\mathbf{q}^{t+1} \leftarrow \mathbf{q}^t + \mathrm{Diag}(1./\boldsymbol{\tau}_r^t)(\mathbf{x}^{t+1} - \mathbf{v}^t)$
11: $\mathbf{s}^{t+1} \leftarrow \mathbf{s}^t + \mathrm{Diag}(1./\boldsymbol{\tau}_p^t)(\mathbf{z}^{t+1} - \mathbf{A}\mathbf{v}^t)$
12: Compute $\mathbf{v}^{t+1}$ from (50)
13:
14: {Compute the gradient terms}
15: $\boldsymbol{\tau}_x^{t+1} \leftarrow \boldsymbol{\tau}_r^t.g_x'(\mathbf{r}^t, \boldsymbol{\tau}_r^t)$, $\boldsymbol{\tau}_z^{t+1} \leftarrow \boldsymbol{\tau}_p^t.g_z'(\mathbf{p}^t, \boldsymbol{\tau}_p^t)$
16: $\overline{\boldsymbol{\tau}}_p^{t+1} \leftarrow \mathbf{S}\boldsymbol{\tau}_x^{t+1}$
17: $\boldsymbol{\tau}_s^{t+1} \leftarrow (1 - \boldsymbol{\tau}_z^{t+1}./\overline{\boldsymbol{\tau}}_p^{t+1})./\overline{\boldsymbol{\tau}}_p^{t+1}$
18: $\overline{\boldsymbol{\tau}}_r^{t+1} \leftarrow \mathbf{1}./(\mathbf{S}^{\mathsf{T}}\boldsymbol{\tau}_s^{t+1})$
19:
20: {Update the linearization}
21: Select a damping parameter $\theta^t \in [0, 1]$
22: $\mathbf{1}./\boldsymbol{\tau}_r^{t+1} \leftarrow \theta^t\mathbf{1}./\overline{\boldsymbol{\tau}}_r^{t+1} + (1-\theta^t)\mathbf{1}./\boldsymbol{\tau}_r^t$
23: $\mathbf{1}./\boldsymbol{\tau}_p^{t+1} \leftarrow \theta^t\mathbf{1}./\overline{\boldsymbol{\tau}}_p^{t+1} + (1-\theta^t)\mathbf{1}./\boldsymbol{\tau}_p^t$
24: **until** Terminated

of these distributions as

$$\mathbf{x}^{t+1} = \mathbb{E}(\mathbf{x}|b_x^{t+1}) \triangleq g_x(\mathbf{r}^t, \boldsymbol{\tau}_r^t), \tag{52a}$$
$$\mathbf{z}^{t+1} = \mathbb{E}(\mathbf{z}|b_z^{t+1}) \triangleq g_z(\mathbf{p}^t, \boldsymbol{\tau}_p^t), \tag{52b}$$

as reflected in line 9 of Algorithm 3. For separable f_x and f_z, we note that the computations in (52) can be performed in a componentwise, scalar manner, e.g.,

$$\begin{aligned} x_j^{t+1} &= [g_x(\mathbf{r}^t, \boldsymbol{\tau}_r^t)]_j \triangleq g_{x_j}(r_j^t, \tau_{r_j}^t) \\ &= \frac{\int_{\mathbb{R}} x \exp\big(-f_{x_j}(x) - \frac{1}{2\tau_{r_j}^t}(x - r_j^t)^2\big)\,\mathrm{d}x}{\int_{\mathbb{R}} \exp\big(-f_{x_j}(x) - \frac{1}{2\tau_{r_j}^t}(x - r_j^t)^2\big)\,\mathrm{d}x}, \end{aligned} \tag{53}$$

$$\begin{aligned} z_i^{t+1} &= [g_z(\mathbf{p}^t, \boldsymbol{\tau}_p^t)]_j \triangleq g_{z_i}(p_i^t, \tau_{p_i}^t) \\ &= \frac{\int_{\mathbb{R}} z \exp\big(-f_{z_i}(z) - \frac{1}{2\tau_{p_i}^t}(z - p_i^t)^2\big)\,\mathrm{d}z}{\int_{\mathbb{R}} \exp\big(-f_{z_i}(z) - \frac{1}{2\tau_{p_i}^t}(z - p_i^t)^2\big)\,\mathrm{d}z}, \end{aligned} \tag{54}$$

Furthermore, the variances of $b_{x_j}^{t+1}$ and $b_{z_i}^{t+1}$ can be computed in a componentwise manner using the derivatives of g_{x_j} and g_{z_i} with respect to their first argument [21], i.e.,

$$\boldsymbol{\tau}_x^{t+1} \triangleq \mathrm{var}(\mathbf{x}|b_x^{t+1}) = \boldsymbol{\tau}_r^t.g_x'(\mathbf{r}^t, \boldsymbol{\tau}_r^t), \tag{55a}$$
$$\boldsymbol{\tau}_z^{t+1} \triangleq \mathrm{var}(\mathbf{z}|b_z^{t+1}) = \boldsymbol{\tau}_p^t.g_z'(\mathbf{p}^t, \boldsymbol{\tau}_p^t), \tag{55b}$$

as reflected in line 15 of Algorithm 3. That is,

$$\tau_{x_j}^{t+1} = [\boldsymbol{\tau}_r^t.g_x'(\mathbf{r}^t, \boldsymbol{\tau}_r^t)]_j = \tau_{r_j}^t \frac{\partial g_{x_j}(r_j^t, \tau_{r_j}^t)}{\partial r_j}. \tag{56}$$

We use these general scalar estimation functions g_x and g_z since it will allow us later to consider a similar algorithm for the MAP estimation problem (5).

Interestingly, the ADMM-GAMP algorithm has close similarities to the sum-product version of the original GAMP algorithm from [21], as we will discuss in Section VIII. For example, the sum-product version of the GAMP algorithm uses the same estimation functions g_x and g_z from (52), which we will refer to as the *MMSE estimation functions*.

D. Computational Cost

While we will demonstrate below that ADMM-GAMP offers improved convergence stability relative to the GAMP algorithm of [21], it is important to point out that the computational cost of ADMM-GAMP may be somewhat larger: One of the main attractive features of GAMP and other first order methods, is that each iteration requires only matrix-vector multiplies by by $\mathbf{A}$ and $\mathbf{A}^{\mathsf{T}}$. Each such multiplication will have complexity $O(mn)$ in the most general case, and may be smaller for structured transforms such as filters, FFTs, or sparse matrices.

In contrast, ADMM-GAMP requires a least-squares (LS) minimization (50) in each iteration. Exact evaluation of the minimization will have a cost of $O(n^2m)$ – a cost not incurred in GAMP or most other first-order methods. As is done ADMM [9] – and in the simulations below – the minimization can be performed approximately via conjugate gradient (CG) [45]. Conjugate gradient also requires repeated matrix-vector multiplies by $\mathbf{A}$ and $\mathbf{A}^{\mathsf{T}}$, but will require K such matrix-vector multiplies where K is the number of CG iterations. In the simulations below, we will use $K = 3$, thus increasing the per iteration cost of ADMM-GAMP by a factor of approximately 3 relative to standard GAMP.

The other computations in each iteration of ADMM-GAMP are typically smaller than the LS minimization and are similar to those performed in GAMP. For example, similar to GAMP, each iteration requires evaluation of the estimation functions $g_x(\cdot)$ and $g_z(\cdot)$. These can be performed as n and m componentwise scalar functions given in (53) and (54). For certain penalty functions, such as Bernoulli-Gaussians, these will have closed-form expressions; otherwise, they will need to be evaluated via numerical integration. In either case, the componentwise cost does not grow with the dimension, so the per iteration cost of evaluating the estimation functions is $O(m + n)$ and are typically not dominant for large m and n.

VI. ADMM-GAMP for MAP Estimation

A. ADMM Inner Loop

For the posterior density $p(\mathbf{x}|\mathbf{y})$ in (2), the MAP estimates of the vector $\mathbf{x}$ and its transform $\mathbf{z} = \mathbf{A}\mathbf{x}$ are given by the constrained optimization

$$(\widehat{\mathbf{x}}, \widehat{\mathbf{z}}) \triangleq \arg\min_{\mathbf{x},\mathbf{z}} J(\mathbf{x}, \mathbf{z}) \text{ s.t. } \mathbf{z} = \mathbf{A}\mathbf{x}, \tag{57}$$

where $J(\mathbf{x}, \mathbf{z})$ is the objective function

$$J(\mathbf{x}, \mathbf{z}) \triangleq f_x(\mathbf{x}) + f_z(\mathbf{z}). \tag{58}$$

We will show that, with appropriate selection of the estimation functions, g_x and g_z, the inner loop of Algorithm 3 can be used as an ADMM method for solving (57).

As before, we replace the constraint $\mathbf{z} = \mathbf{A}\mathbf{x}$ in the optimization (57) with two constraints: $\mathbf{x} = \mathbf{v}$ and $\mathbf{z} = \mathbf{A}\mathbf{v}$. We then define the augmented Lagrangian

$$\begin{aligned} &L(\mathbf{x}, \mathbf{z}, \mathbf{s}, \mathbf{q}, \mathbf{v}; \boldsymbol{\tau}_p, \boldsymbol{\tau}_r) \\ &\quad \triangleq f_x(\mathbf{x}) + f_z(\mathbf{z}) + \mathbf{q}^{\mathsf{T}}(\mathbf{x} - \mathbf{v}) + \mathbf{s}^{\mathsf{T}}(\mathbf{z} - \mathbf{A}\mathbf{v}) \\ &\qquad + \tfrac{1}{2}\|\mathbf{x} - \mathbf{v}\|^2_{\boldsymbol{\tau}_r} + \tfrac{1}{2}\|\mathbf{z} - \mathbf{A}\mathbf{v}\|^2_{\boldsymbol{\tau}_p}. \end{aligned} \tag{59}$$

The ADMM recursions (40) for this augmented Lagrangian are then given by

$$(\mathbf{x}^{t+1}, \mathbf{z}^{t+1}) = \arg\min_{\mathbf{x},\mathbf{z}} L(\mathbf{x}, \mathbf{z}, \mathbf{s}^t, \mathbf{q}^t, \mathbf{v}^t; \boldsymbol{\tau}_p, \boldsymbol{\tau}_r), \tag{60a}$$

$$\mathbf{s}^{t+1} = \mathbf{s}^t + \mathrm{Diag}(\mathbf{1}./\boldsymbol{\tau}_p)\big(\mathbf{z}^{t+1} - \mathbf{A}\mathbf{v}^t\big), \tag{60b}$$

$$\mathbf{q}^{t+1} = \mathbf{q}^t + \mathrm{Diag}(\mathbf{1}./\boldsymbol{\tau}_r)\big(\mathbf{x}^{t+1} - \mathbf{v}^t\big), \tag{60c}$$

$$\mathbf{v}^{t+1} = \arg\min_{\mathbf{v}} L(\mathbf{x}^{t+1}, \mathbf{z}^{t+1}, \mathbf{s}^{t+1}, \mathbf{q}^{t+1}, \mathbf{v}; \boldsymbol{\tau}_p, \boldsymbol{\tau}_r). \tag{60d}$$

To perform the minimization in (60a), first consider the minimization over $\mathbf{x}$. Eliminating terms that do not depend on $\mathbf{x}$, we obtain

$$\begin{aligned} \mathbf{x}^{t+1} &= \arg\min_{\mathbf{x}} f_x(\mathbf{x}) + \mathbf{q}^{\mathsf{T}}\mathbf{x} + \tfrac{1}{2}\|\mathbf{x} - \mathbf{v}\|^2_{\boldsymbol{\tau}_r} \\ &= \arg\min_{\mathbf{x}} f_x(\mathbf{x}) + \tfrac{1}{2}\|\mathbf{x} + \boldsymbol{\tau}_r.\mathbf{q} - \mathbf{v}\|^2_{\boldsymbol{\tau}_r}. \end{aligned} \tag{61}$$

Similarly, the minimization over $\mathbf{z}$ reduces to

$$\begin{aligned} \mathbf{z}^{t+1} &= \arg\min_{\mathbf{z}} f_z(\mathbf{z}) + \mathbf{s}^{\mathsf{T}}\mathbf{z} + \tfrac{1}{2}\|\mathbf{z} - \mathbf{A}\mathbf{v}\|^2_{\boldsymbol{\tau}_p} \\ &= \arg\min_{\mathbf{z}} f_z(\mathbf{z}) + \tfrac{1}{2}\|\mathbf{z} + \boldsymbol{\tau}_p.\mathbf{s} - \mathbf{A}\mathbf{v}\|^2_{\boldsymbol{\tau}_p}. \end{aligned} \tag{62}$$

Hence, if we define the estimation functions

$$g_x(\mathbf{r}, \boldsymbol{\tau}_r) \triangleq \arg\min_{\mathbf{x}} \Big[f_x(\mathbf{x}) + \tfrac{1}{2}\|\mathbf{x} - \mathbf{r}\|^2_{\boldsymbol{\tau}_r} \Big], \tag{63a}$$

$$g_z(\mathbf{p}, \boldsymbol{\tau}_p) \triangleq \arg\min_{\mathbf{z}} \Big[f_x(\mathbf{x}) + \tfrac{1}{2}\|\mathbf{z} - \mathbf{p}\|^2_{\boldsymbol{\tau}_p} \Big], \tag{63b}$$

then we can rewrite (61) and (62) as

$$\mathbf{x}^{t+1} = g_x(\mathbf{r}^t, \boldsymbol{\tau}_r), \quad \mathbf{z}^{t+1} = g_z(\mathbf{p}^t, \boldsymbol{\tau}_p), \tag{64}$$

for $\mathbf{r}^t$ and $\mathbf{p}^t$ defined in (49). Also, the minimization over $\mathbf{v}$ in (60d) can again be cast as the least-squares problem (50).

We see that equations (49), (50), (60b), (60c) and (64) are precisely the updates in the ADMM inner-loop of Algorithm 3. Therefore, for fixed penalty terms $\boldsymbol{\tau}_r$ and $\boldsymbol{\tau}_p$, the inner loop of the ADMM-GAMP algorithm with the estimation functions (63) is precisely an ADMM algorithm for the MAP estimation problem (57).

The functions in (63) are the standard "proximal" operators used in many implementations of ADMM and related optimization algorithms [9]. These functions also appear in the max-sum version of GAMP from [21], which is used for MAP estimation. Thus, we will refer to (63) as the *MAP estimation functions*.

B. Hardening Limit of the LSL-BFE

The above discussion shows that, with the MAP estimation functions (63), the ADMM-GAMP outputs $(\mathbf{x}^t, \mathbf{z}^t)$ can be interpreted as estimates of the MAP solution from (57). How then do we interpret the related terms $(\boldsymbol{\tau}_x^t, \boldsymbol{\tau}_z^t)$? In the inference (i.e., MMSE) problem from Section V, the components of $\boldsymbol{\tau}_x^t$ and $\boldsymbol{\tau}_z^t$ are estimates of the variances of the marginal posteriors. Below, we use a hardening argument to show that, in the MAP problem, $(\boldsymbol{\tau}_x^t, \boldsymbol{\tau}_z^t)$ can be interpreted as estimates of the local curvature of the MAP objective (58).

To be precise, let us define the *marginal minimization functions*

$$\phi_{x_j}(x_j) \triangleq \min_{\mathbf{x}\setminus x_j} J(\mathbf{x}, \mathbf{A}\mathbf{x}), \tag{65a}$$

$$\phi_{z_i}(z_i) \triangleq \min_{\mathbf{x}: z_i = [\mathbf{A}\mathbf{x}]_i} J(\mathbf{x}, \mathbf{A}\mathbf{x}), \tag{65b}$$

where the minimizations are over the vector $\mathbf{x}$, holding either x_j or $z_i \triangleq [\mathbf{A}\mathbf{x}]_i$ fixed. Note that, if one can compute these marginal minimization functions, then one can compute the components of the MAP estimates from (57) via

$$\widehat{x}_j = \arg\min_{x_j} \phi_{x_j}(x_j), \quad \widehat{z}_i = \arg\min_{z_i} \phi_{z_i}(z_i). \tag{66}$$

However, the marginal minimization functions provide not only componentwise objectives for the MAP optimization (57), but also the sensitivity of those objectives.

We will see that ADMM-GAMP provides estimates of the marginal minimization functions, in addition to estimates of the MAP solution in (57). Perhaps the easiest way to see this is through a standard "hardening" analysis, which is also used to understand how max-sum loopy belief propagation can be viewed as a limit of sum-product loopy BP; see, for example, [46], [47]. Specifically, combining (2) with Laplace's Principle from large deviations [48], and assuming suitable continuity conditions, the marginal minimization functions (65) are given by (up to a constant factor)

$$\begin{aligned} \phi_{x_j}(x_j) &= -\lim_{T\to 0} T \ln p_{x_j}(x_j; T), \\ \phi_{z_i}(z_i) &= -\lim_{T\to 0} T \ln p_{z_i}(z_i; T), \end{aligned}$$

where $p_{x_j}(x_j; T)$ and $p_{z_i}(z_i; T)$ are the marginal densities for the *scaled* joint density

$$p(\mathbf{x}; T) \triangleq \frac{1}{Z} \exp\left[-\frac{1}{T}\Big(f_x(\mathbf{x}) + f_z(\mathbf{A}\mathbf{x}) \Big) \right].$$

Note that, for any $T > 0$, we can estimate the marginal posteriors $p_{x_j}(x_j; T)$ and $p_{z_i}(z_i; T)$ using the LSL-BFE optimization from Section V. That is, we can use the estimate

$$\phi_{x_j}(x_j) \approx \widehat{\phi}_{x_j}(x_j) \triangleq -\lim_{T\to 0} T \ln \widehat{b}_{x_j}(x_j; T), \tag{67a}$$

$$\phi_{z_i}(z_i) \approx \widehat{\phi}_{z_i}(z_i) \triangleq -\lim_{T\to 0} T \ln \widehat{b}_{z_i}(z_i; T), \tag{67b}$$

where $\widehat{b}_{x_j}(x_j; T)$ and $\widehat{b}_{z_i}(z_i; T)$ are the belief estimates computed via the LSL-BFE optimization under the scaled penalties

$$f_x(\mathbf{x}; T) \triangleq f_x(\mathbf{x})/T, \quad f_z(\mathbf{z}; T) \triangleq f_z(\mathbf{z})/T. \tag{68}$$

In statistical physics, the parameter T has the interpretation of temperature, and the limit $T \to 0$ corresponds to a "cooling"

of the system. In inference problems, the cooling has the effect of concentrating the distributions about their maxima.

A large-deviations analysis in Appendix B shows that, if we use ADMM-GAMP with the MMSE estimation functions (52) with the scaled functions (68), then at iteration t the limits in (67) are given by

$$\begin{aligned}\widehat{\phi}^t_{x_j}(x_j) &= -\lim_{T\to 0} T \ln b^t_{x_j}(x_j;T),\\ &= f_{x_j}(x_j) + \tfrac{1}{2\tau^t_{r_j}}(x_j - r^t_j)^2 \qquad (69a)\\ \widehat{\phi}^t_{z_i}(z_i) &= -\lim_{T\to 0} T \ln b^t_{z_i}(z_i;T)\\ &= f_{z_i}(z_i) + \tfrac{1}{2\tau^t_{p_i}}(z_i - p^t_i)^2, \qquad (69b)\end{aligned}$$

where the parameters r^t_j, p^t_i, $\tau^t_{r_j}$, and $\tau^t_{p_i}$ are the outputs of ADMM-GAMP under the MAP estimation functions (63). In this sense, ADMM-GAMP under the MAP estimation function can be seen as a limiting case of ADMM-GAMP under the MMSE estimation functions. Hence, according to (67), MAP ADMM-GAMP can be used to compute estimates (69) of the marginal minimization functions (65). Furthermore, according to (63) and (64), $\mathbf{x}^{t+1}$ and $\mathbf{z}^{t+1}$ are the minima of these functions

$$\widehat{x}^{t+1}_j = \arg\min_{x_j} \widehat{\phi}^t_{x_j}(x_j), \quad \widehat{z}^{t+1}_i = \arg\min_{z_i} \widehat{\phi}^t_{z_i}(z_i),$$

as one would expect from (66).

Finally, it can be shown (see (111)) that, for the MAP estimation functions (63), the outputs of line 15 in Algorithm 3 take the form

$$\tau_{x_j} = \tau_{r_j} g'_{x_j}(r_j, \tau_{p_i}) = \frac{\tau_{r_j}}{1 + \tau_{r_j} f''_{x_j}(\widehat{x}_j)}, \qquad (70a)$$

$$\tau_{z_i} = \tau_{p_i} g'_{z_i}(p_i, \tau_{p_i}) = \frac{\tau_{p_i}}{1 + \tau_{p_i} f''_{z_i}(\widehat{z}_i)}. \qquad (70b)$$

Meanwhile, from (69), we see that

$$\frac{1}{\tau^{t+1}_{x_j}} = \frac{\partial^2 \widehat{\phi}^t_{x_j}(\widehat{x}^{t+1}_j)}{\partial x_j^2}, \quad \frac{1}{\tau^{t+1}_{z_i}} = \frac{\partial^2 \widehat{\phi}^t_{z_i}(\widehat{z}^{t+1}_i)}{\partial z_i^2}. \qquad (71)$$

Therefore, when ADMM-GAMP is used for MAP estimation, the components of $\boldsymbol{\tau}^t_x$ and $\boldsymbol{\tau}^t_z$ can be interpreted as the inverse curvatures of the constrained function $J(\mathbf{x}, \mathbf{z} = \mathbf{A}\mathbf{x})$ in the vicinity of the current estimate $(\widehat{\mathbf{x}}^t, \widehat{\mathbf{z}}^t)$.

Appendix B also show that, in the limit as $T \to 0$, the LSL-BFE optimization (14) decomposes approximately into two decoupled optimizations: The first computes the MAP estimates $(\widehat{\mathbf{x}}, \widehat{\mathbf{z}})$ from (57), and the second computes

$$(\widehat{\boldsymbol{\tau}}_x, \widehat{\boldsymbol{\tau}}_z) \triangleq \arg\min_{\boldsymbol{\tau}_x, \boldsymbol{\tau}_z} J^2(\boldsymbol{\tau}_x, \boldsymbol{\tau}_z, \widehat{\mathbf{x}}, \widehat{\mathbf{z}}), \qquad (72)$$

where

$$\begin{aligned}J^2(\boldsymbol{\tau}_x, \boldsymbol{\tau}_z, \widehat{\mathbf{x}}, \widehat{\mathbf{z}}) \triangleq &\sum_{j=1}^{n} \left[\tau_{x_j} f''_{x_j}(\widehat{x}_j) - \ln(\tau_{x_j})\right]\\ &+ \sum_{i=1}^{m} \left[\tau_{z_i}\left(f''_{z_i}(\widehat{z}_i) + \frac{1}{\tau_{p_i}}\right) + \ln\left(\frac{\tau_{p_i}}{\tau_{z_i}}\right)\right],\end{aligned} \qquad (73)$$

and, as before, $\boldsymbol{\tau}_p \triangleq \mathbf{S}\boldsymbol{\tau}_x$. Since the optimization (72) provides the inverse-curvature estimates in (71), we will refer to it as *curvature optimization*.

VII. Convergence Analysis for Strictly Convex Penalties

A. Fixed Points of ADMM-GAMP

We first characterize the fixed points of ADMM-GAMP, assuming that the algorithm converges.

Theorem 2: At any fixed point of ADMM-GAMP with the MMSE estimation functions (52), the belief pair (b_x, b_z) in (48) is a critical point of the constrained LSL-BFE optimization (14).

Proof: See Appendix C. ∎

Theorem 3: At any fixed point of ADMM-GAMP with the MAP estimation functions (63), the output $(\mathbf{x}, \mathbf{z})$ is a critical point of the constrained MAP optimization (57) and $(\boldsymbol{\tau}_x, \boldsymbol{\tau}_z)$ is a critical point of the optimization (72).

Proof: See Appendix C. ∎

Theorems 2 and 3 show that, if ADMM-GAMP converges, then its limit points will be local minima of either the inference (i.e., MMSE) or MAP problems.

B. Convergence of the ADMM Inner Loop

For the remainder of this section, we will show the convergence of ADMM-GAMP in the special case of convex and smooth penalties f_x and f_z. We begin by analyzing the convergence of the ADMM inner-loop under fixed linearization terms $\boldsymbol{\tau}_r$ and $\boldsymbol{\tau}_p$. It is well-known that, when one applies ADMM to a general optimization problem of the form (38) with convex f and full-rank $\mathbf{B}$, the method will converge [9]. However, in our case, the objective function is the linearized LSL-BFE in (31), which is not necessarily convex, even if the penalty functions f_x and f_z are. The problem is that the variances $\text{var}(\mathbf{x}|b_x)$ and $\text{var}(\mathbf{z}|b_z)$ are not convex functions of the densities b_x and b_z (in fact, they are concave). We thus need a separate proof.

We will prove convergence under the following assumption.

Assumption 2: For fixed $\boldsymbol{\tau}_r$ and $\boldsymbol{\tau}_p$, the estimation functions $g_x(\mathbf{r}, \boldsymbol{\tau}_r)$ and $g_z(\mathbf{p}, \boldsymbol{\tau}_p)$ are separable in $\mathbf{r}$ and $\mathbf{p}$ in that

$$\begin{aligned}g_x(\mathbf{r}, \boldsymbol{\tau}_r) &= \big(g_{x_1}(r_1, \boldsymbol{\tau}_r), \cdots, g_{x_n}(r_n, \boldsymbol{\tau}_r)\big),\\ g_z(\mathbf{p}, \boldsymbol{\tau}_p) &= \big(g_{z_1}(p_1, \boldsymbol{\tau}_p), \cdots, g_{z_m}(p_m, \boldsymbol{\tau}_p)\big)\end{aligned}$$

for scalar function g_{x_j} and g_{z_i}. In addition, these scalar functions have, with respect to their first arguments, continuous first derivatives g'_{x_j} and g'_{z_i} satisfying

$$\epsilon \le g'_{x_j}(r_j, \boldsymbol{\tau}_r) \le 1 - \epsilon, \quad \epsilon \le g'_{z_i}(p_i, \boldsymbol{\tau}_p) \le 1 - \epsilon. \qquad (74)$$

for some constant $\epsilon \in (0, 0.5]$.

The assumption requires that the estimation functions are strictly increasing contractions. Importantly, the following lemma shows that this assumption holds when the penalty functions are smooth and convex.

Lemma 1: Suppose that f_x and f_z are strictly convex, separable functions, in that they are of the form (4), where the components have continuous second derivatives such that

$$A \le f''_{x_j}(x_j) \le B \ \forall x_j, \quad A \le f''_{z_i}(z_i) \le B \ \forall z_i, \qquad (75)$$

for some $0 < A \leq B < \infty$. Then, both the MMSE estimation functions in (52) and the MAP estimation functions in (63) satisfy Assumption 2 for any $\boldsymbol{\tau}_r, \boldsymbol{\tau}_p > 0$.

Proof: See Appendix D. ∎

We now have the following convergence result.

Theorem 4: Consider Algorithm 3 with only ADMM updates (i.e., $\theta^t = 0$ for all t), so that the linearization terms remain constant, i.e.,

$$\boldsymbol{\tau}_p^t = \boldsymbol{\tau}_p \text{ and } \boldsymbol{\tau}_r^t = \boldsymbol{\tau}_r \ \forall t,$$

for some vectors $\boldsymbol{\tau}_p$ and $\boldsymbol{\tau}_r$. Then, if the estimation functions satisfy Assumption 2, the algorithm converges to a unique fixed point at a linear rate of convergence of $1 - \epsilon$.

Proof: See Appendix E. ∎

C. Outer Loop Convergence: MMSE Case

Theorem 4 shows that, with the MMSE estimation functions (52) and strictly convex penalties, the ADMM inner loop of Algorithm 3 converges. We next consider the convergence of the outer loop, Algorithm 2, assuming that the inner minimization (i.e., line 5 of Algorithm 2) is computed exactly.

Theorem 5: Suppose that the functions f_x and f_z satisfy the assumptions in Lemma 1 and the matrix $\mathbf{S}$ has positive components (i.e., $S_{ij} = |A_{ij}|^2 > 0 \ \forall ij$). Then, there exists a $\overline{\theta}$ such that, if $\theta^k < \overline{\theta}$, the sequence of belief estimates b^k generated by Algorithm 2 yields a monotonically non-increasing LSL-BFE, i.e.,

$$J(b_x^{k+1}, b_z^{k+1}) \leq J(b_x^k, b_z^k).$$

Proof: See Appendix F. ∎

Together, Theorems 4 and 5 demonstrate that ADMM-GAMP will converge under an infinitely slow damping schedule. Specifically, we select iterations $t_1 < t_2 < \cdots$ that are infinitely far apart. Then, for all t between each t_k and t_{k+1}, we set $\theta^t = 0$ so that the ADMM inner-loop is run to completion, and at each $t = t_k$, we select θ^t to be a small positive value.

It is of course impossible to use an infinite number of inner-loop iterations in practice. Fortunately, our numerical experiments in Section IX suggest that a fixed number of inner-loop iterations is sufficient.

D. Outer Loop Convergence: MAP Case

We can prove a stronger convergence result for ADMM-GAMP under the MAP estimation functions (63), if we make two additional assumptions. Recall from Theorem 4 that, if we set $\theta^t = 0 \ \forall t$, then the linearization parameters $\boldsymbol{\tau}_r^t$ and $\boldsymbol{\tau}_p^t$ will remain constant with t and the algorithm will converge to some fixed point. Our first assumption is that we begin the algorithm at one such fixed point. That is, we suppose that the time $t = 0$ versions of

$$\mathbf{x}^t, \ \mathbf{z}^t, \ \mathbf{r}^t, \ \mathbf{p}^t, \ \mathbf{q}^t, \ \mathbf{s}^t, \ \mathbf{v}^t \tag{76}$$

are fixed points of lines 7 through 12 in Algorithm 3. Our second assumption is that we replace the $\boldsymbol{\tau}_s^{t+1}$ update in line 17 with

$$\boldsymbol{\tau}_s^{t+1} \leftarrow (\mathbf{1} - \boldsymbol{\tau}_z^{t+1}./\boldsymbol{\tau}_p^t)./\boldsymbol{\tau}_p^t. \tag{77}$$

That is, we use $\boldsymbol{\tau}_p^t$ instead of $\overline{\boldsymbol{\tau}}_p^{t+1}$. Under these two additional assumptions, we can prove the following.

Theorem 6: Consider ADMM-GAMP, Algorithm 3, run under the MAP estimation functions (63), with penalty functions f_x and f_z satisfying the assumptions of Lemma 1. Suppose that the initialization (76) is a fixed point of lines 7 through 12, and that line 17 is replaced by (77). Then, if $\theta^t = 1$ for all t,

1) Even though $\boldsymbol{\tau}_r^t$ and $\boldsymbol{\tau}_p^t$ may change with t, the variables in (76) will remain constant. That is, for all t,
$$\mathbf{x}^t = \mathbf{x}^0, \ \mathbf{z}^t = \mathbf{z}^0, \ \mathbf{r}^t = \mathbf{r}^0, \ \mathbf{p}^t = \mathbf{p}^0,$$
$$\mathbf{q}^t = \mathbf{q}^0, \ \mathbf{s}^t = \mathbf{s}^0, \ \mathbf{v}^t = \mathbf{v}^0. \tag{78}$$
Moreover, the variables $(\mathbf{x}^0, \mathbf{z}^0)$ are the global minima of the MAP estimation problem (57).
2) The linearization parameters $\boldsymbol{\tau}_x^t$ and $\boldsymbol{\tau}_z^t$ converge to unique global minima of the curvature optimization G.

Proof: See Appendix G. ∎

The result shows that, in principle, we can solve the MAP estimation problem by first running the ADMM inner loop to convergence with arbitrary positive linearization terms $\boldsymbol{\tau}_r$ and $\boldsymbol{\tau}_p$. Then, we could turn on the outer loop updates, thus driving $\boldsymbol{\tau}_x^t$ and $\boldsymbol{\tau}_z^t$ to the minima of the curvature optimization problem (71). Of course, in practice, one cannot do this perfectly, since the ADMM inner loop must be terminated at some finite number of iterations. Also, it is possible that, by letting the variance terms adapt (at least slowly) before the inner loop fully converges, the convergence speed of the inner loop can be improved. In fact, this is our empirical experience, although we have no proof.

It is important to point out that the MAP convergence proof requires a slightly modified variance update given in (77). This update may actually be preferable for the MMSE case as well, however, further analysis would be required. Indeed, while we have demonstrated one variance update with provable convergence, finding the best variance update method is a still an open question.

VIII. Relationship of ADMM-GAMP to GAMP

There are two key differences between the proposed ADMM-GAMP algorithm and the original sum-product GAMP algorithm from [21], reproduced for convenience in Algorithm 4 (with the variance updates indented for visual clarity).

1) The ADMM-GAMP algorithm uses two additional variables: a dual variable $\mathbf{q}^t$, and an auxiliary variable $\mathbf{v}^t$ that is updated via the least-squares optimization (50), that are not present in the original GAMP algorithm.
2) ADMM-GAMP uses an alternating schedule of mean and (possibly damped) variance updates, whereas GAMP uses interleaved mean and variance updates.

Below, we describe these differences in more detail.

A. Sum-Product GAMP via Stale, Linearized ADMM

One way to understand the differences between ADMM-GAMP and the original GAMP is as follows: ADMM-GAMP results from minimizing the linearized LSL-BFE via ADMM under the splitting rule "$\mathbb{E}(\mathbf{z}|b_z) = \mathbf{Av}$

Algorithm 4 Original GAMP

Require: Matrix $\mathbf{A}$ and estimation functions g_x and g_z.
1: $\mathbf{S} \leftarrow \mathbf{A}.\mathbf{A}$ (componentwise square)
2: Initialize $\mathbf{x}^0$, $\boldsymbol{\tau}_x^0$
3: $\mathbf{s}^0 \leftarrow 0$
4: $t \leftarrow 0$
5: **repeat**
6: $\quad \boldsymbol{\tau}_p^t \leftarrow \mathbf{S}\boldsymbol{\tau}_x^t$
7: $\quad \mathbf{p}^t \leftarrow \mathbf{A}\mathbf{x}^t - \boldsymbol{\tau}_p^t.\mathbf{s}^{t-1}$
8: $\quad \boldsymbol{\tau}_z^t \leftarrow \boldsymbol{\tau}_p^t.g_z'(\mathbf{p}^t, \boldsymbol{\tau}_p^t)$
9: $\quad \mathbf{z}^t \leftarrow g_z(\mathbf{p}^t, \boldsymbol{\tau}_p^t)$
10: $\quad \boldsymbol{\tau}_s^t \leftarrow (\mathbf{1} - \boldsymbol{\tau}_z^t./\boldsymbol{\tau}_p^t)./\boldsymbol{\tau}_p^t$
11: $\quad \mathbf{s}^t \leftarrow (\mathbf{z}^t - \mathbf{p}^t)./\boldsymbol{\tau}_p^t$
12: $\quad \boldsymbol{\tau}_r^t \leftarrow \mathbf{1}./(\mathbf{S}^{\mathsf{T}}\boldsymbol{\tau}_s^t)$
13: $\quad \mathbf{r}^t \leftarrow \mathbf{x}^t + \mathrm{Diag}(\boldsymbol{\tau}_r^t)\mathbf{A}^{\mathsf{T}}\mathbf{s}^t$
14: $\quad \boldsymbol{\tau}_x^{t+1} \leftarrow \boldsymbol{\tau}_r^t.g_x'(\mathbf{r}^t, \boldsymbol{\tau}_r^t)$
15: $\quad \mathbf{x}^{t+1} \leftarrow g_x(\mathbf{r}^t, \boldsymbol{\tau}_r^t)$
16: **until** Terminated

and $\mathbb{E}(\mathbf{x}|b_x) = \mathbf{v}$" (as described in Section V-B), whereas the original GAMP uses *stale, linearized* ADMM under the conventional[1] splitting rule "$\mathbb{E}(\mathbf{z}|b_z) = \mathbf{A}\mathbb{E}(\mathbf{x}|b_x)$." Both use the same iterative LSL-BFE linearization strategy described in Section IV-D.

We can derive the mean updates in the original GAMP using the augmented Lagrangian

$$\begin{aligned} &L(b_x, b_z, \mathbf{s}; \boldsymbol{\tau}_p) \\ &\quad \triangleq J(b_x, b_z, \boldsymbol{\tau}_r, \boldsymbol{\tau}_p) + \mathbf{s}^{\mathsf{T}}\big(\mathbb{E}(\mathbf{z}|b_z) - \mathbf{A}\mathbb{E}(\mathbf{x}|b_x)\big) \\ &\qquad + \tfrac{1}{2}\|\mathbb{E}(\mathbf{z}|b_z) - \mathbf{A}\mathbb{E}(\mathbf{x}|b_x)\|_{\boldsymbol{\tau}_p}^2, \end{aligned} \tag{79}$$

for the J defined in (31) and *stale, linearized* ADMM:

$$\begin{aligned} b_x^{t+1} &= \arg\min_{b_x} L(b_x, b_z^t, \mathbf{s}^{t-1}; \boldsymbol{\tau}_p) + \tfrac{1}{2}\big(\mathbb{E}(\mathbf{x}|b_x) - \mathbb{E}(\mathbf{x}|b_x^t)\big)^{\mathsf{T}} \\ &\quad \times \big(\mathbf{D}_{\boldsymbol{\tau}_r} - \mathbf{A}^{\mathsf{T}}\mathbf{D}_{\boldsymbol{\tau}_p}\mathbf{A}\big)\big(\mathbb{E}(\mathbf{x}|b_x) - \mathbb{E}(\mathbf{x}|b_x^t)\big), \end{aligned} \tag{80a}$$

$$b_z^{t+1} = \arg\min_{b_z} L(b_x^{t+1}, b_z, \mathbf{s}^t; \boldsymbol{\tau}_p), \tag{80b}$$

$$\mathbf{s}^{t+1} = \mathbf{s}^t + \mathbf{D}_{\boldsymbol{\tau}_p}\big(\mathbb{E}(\mathbf{z}|b_z^{t+1}) - \mathbf{A}\mathbb{E}(\mathbf{x}|b_x^{t+1})\big), \tag{80c}$$

where $\mathbf{D}_{\boldsymbol{\tau}} \triangleq \mathrm{Diag}(\mathbf{1}./\boldsymbol{\tau})$. Note the addition of a "linearization" term in (80a) to decouple the minimization. The resulting approach goes by several names: linearized ADMM [49, Sec. 4.4.2], split inexact Uzawa [10], and primal-dual hybrid gradient (PDHG) [10]. Note also the use of the "stale" dual estimate $\mathbf{s}^{t-1}$ in (80a), as opposed to the most recent dual estimate $\mathbf{s}^t$. In the context of PDHG, this stale update is known as Arrow-Hurwicz [10]. In Appendix H, we show that the recursion (80) yields the mean updates in the original sum-product GAMP algorithm (i.e., the non-indented lines in Algorithm 4).

Regarding the variance updates of the original sum-product GAMP algorithm (i.e., the indented lines in Algorithm 4), a visual inspection shows that they match the non-damped ADMM-GAMP "gradient" updates (i.e., lines 15-18 of Algorithm 3 under $\theta^t = 1$), except for one small difference: in the original sum-product GAMP, the update of $\boldsymbol{\tau}_s$ uses the same version of $\boldsymbol{\tau}_p$ used by the $\boldsymbol{\tau}_z$ update, whereas in ADMM-GAMP, the update of $\boldsymbol{\tau}_s$ uses a more recent version of $\boldsymbol{\tau}_p$.

[1]See, e.g., [9, Sec. 3.1].

B. Recovering GAMP From ADMM-GAMP

We now show that the mean-updates of the original sum-product GAMP can be recovered by approximating the mean-updates of ADMM-GAMP. For simplicity, we suppress the t index on the variance terms.

At any critical point of Algorithm 3, we must have $\mathbf{q}^t = -\mathbf{A}^{\mathsf{T}}\mathbf{s}^t$ and $\mathbf{z}^t = \mathbf{A}\mathbf{x}^t$, as shown in (108). If we substitute these two constraints into the $\mathbf{v}$-update objective in (50), we obtain

$$\begin{aligned} &\|\mathbf{z}^t + \boldsymbol{\tau}_p.\mathbf{s}^t - \mathbf{A}\mathbf{v}\|_{\boldsymbol{\tau}_p}^2 + \|\mathbf{x}^t + \boldsymbol{\tau}_r.\mathbf{q}^t - \mathbf{v}\|_{\boldsymbol{\tau}_r}^2 \\ &\quad = \|\mathbf{A}(\mathbf{x}^t - \mathbf{v}) + \boldsymbol{\tau}_p.\mathbf{s}^t\|_{\boldsymbol{\tau}_p}^2 + \|\mathbf{x}^t - \mathbf{v} - \mathrm{Diag}(\boldsymbol{\tau}_r)\mathbf{A}^{\mathsf{T}}\mathbf{s}^t\|_{\boldsymbol{\tau}_r}^2. \end{aligned}$$

It can be verified that the minimum for this function occurs at $\mathbf{v} = \mathbf{x}^t$. So, if we substitute $\mathbf{v}^t = \mathbf{x}^t$ and $\mathbf{q}^t = -\mathbf{A}^{\mathsf{T}}\mathbf{s}^t$ into the mean updates in Algorithm 3, we obtain

$$\begin{aligned} \mathbf{x}^{t+1} &= g_x(\mathbf{r}^t, \boldsymbol{\tau}_r), \\ \mathbf{z}^{t+1} &= g_z(\mathbf{p}^t, \boldsymbol{\tau}_p), \\ \mathbf{s}^{t+1} &= \mathbf{s}^t + \mathrm{Diag}(\mathbf{1}./\boldsymbol{\tau}_p)(\mathbf{z}^{t+1} - \mathbf{A}\mathbf{x}^t), \\ \mathbf{r}^{t+1} &= \mathbf{x}^{t+1} + \mathrm{Diag}(\boldsymbol{\tau}_r)\mathbf{A}^{\mathsf{T}}\mathbf{s}^{t+1}, \\ \mathbf{p}^{t+1} &= \mathbf{A}\mathbf{x}^{t+1} - \boldsymbol{\tau}_p.\mathbf{s}^{t+1}. \end{aligned}$$

Then, substituting the $\mathbf{p}$ update into the $\mathbf{s}$ update, defining $\overline{\mathbf{z}}^t = \mathbf{z}^{t+1}$ and $\overline{\mathbf{s}}^t = \mathbf{s}^{t+1}$, and reordering the steps, we obtain

$$\begin{aligned} \mathbf{p}^t &= \mathbf{A}\mathbf{x}^t - \boldsymbol{\tau}_p\overline{\mathbf{s}}^{t-1}, \\ \overline{\mathbf{z}}^t &= g_z(\mathbf{p}^t, \boldsymbol{\tau}_p), \\ \overline{\mathbf{s}}^t &= \mathrm{Diag}(\mathbf{1}./\boldsymbol{\tau}_p)(\overline{\mathbf{z}}^t - \mathbf{p}^t), \\ \mathbf{r}^t &= \mathbf{x}^t + \mathrm{Diag}(\boldsymbol{\tau}_r)\mathbf{A}^{\mathsf{T}}\overline{\mathbf{s}}^{t-1}, \\ \mathbf{x}^{t+1} &= g_x(\mathbf{r}^t, \boldsymbol{\tau}_r), \end{aligned}$$

which is precisely the GAMP mean-update loop.

IX. Numerical Experiments

We now illustrate the performance of ADMM-GAMP by considering three numerical experiments. While our theoretical results assumed strictly convex penalties, we numerically demonstrate the stability of ADMM-GAMP for the non-convex penalty corresponding to a Bernoulli-Gaussian prior on $\mathbf{x}$, i.e.,

$$p_x(x) = (1 - \rho)\delta(x) + \rho\mathcal{N}(x; 0, 1), \tag{81}$$

where $\rho \in (0, 1]$ is the sparsity ratio and δ is the Dirac delta distribution. In our experiments, we fix the parameters to $n = 1000$ and $\rho = 0.2$, and we numerically compare the normalized MSE

$$\text{NMSE (dB)} \triangleq 10\log_{10}\left(\frac{\|\mathbf{x} - \widehat{\mathbf{x}}\|_2^2}{\|\mathbf{x}\|_2^2}\right)$$

of ADMM-GAMP to four other recovery schemes: the original GAMP method [21]; de-biased LASSO [50]; swept

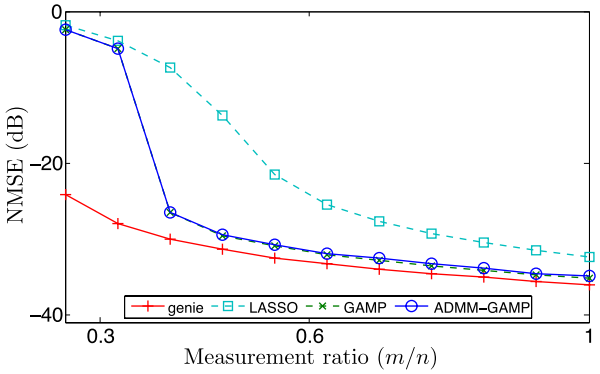

Fig. 2. Average NMSE versus measurement rate m/n when recovering a length $n = 1000$ Bernoulli-Gaussian signal $\mathbf{x}$ from AWGN-corrupted measurements $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$ under i.i.d. $\mathbf{A}$.

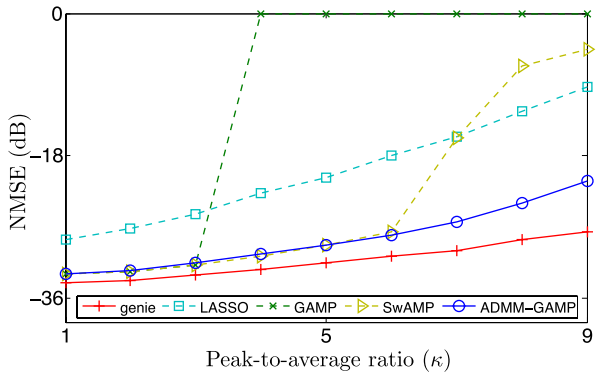

Fig. 3. Average NMSE versus peak-to-average squared-singular-value ratio $\kappa(\mathbf{A})$ when recovering a length $n = 1000$ Bernoulli-Gaussian signal $\mathbf{x}$ from $m = 600$ AWGN-corrupted measurements $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$. Note the superior performance of ADMM-GAMP relative to both the original GAMP and SwAMP, and the proximity of ADMM-GAMP to the support-aware genie.

AMP (SwAMP) [26]; and the support-aware MMSE estimator, labeled "genie." The SwAMP method is identical to original GAMP method but updates only one component of $\mathbf{x}$ at a time – a common technique also used for stabilizing loopy BP. For LASSO, we optimized the regularization parameter λ for best MSE performance. For GAMP, SwAMP, and ADMM-GAMP, we terminated the iterations as soon as $\|\widehat{\mathbf{x}}^t - \widehat{\mathbf{x}}^{t-1}\|_2 / \|\widehat{\mathbf{x}}^{t-1}\|_2 \leq 10^{-4}$ and imposed an upper limit of 200 iterations. In all experiments below, ADMM-GAMP was run with 10 iterations of the inner loop ADMM minimization for each outer loop update. Also, the least-sqaures minimization (50) was performed with 3 conjugate gradient iterations per inner loop iteration, using as a warm start, the output of final value from the previous iteration as the initial condition of the current iteration.

In our first experiment, we consider a standard problem: recover sparse $\mathbf{x}$ from $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$, where $\mathbf{e}$ is AWGN with variance set to achieve an SNR of 30 dB, and where the measurement matrix $\mathbf{A}$ is drawn with i.i.d. $\mathcal{N}(0, 1/m)$ entries. Figure 2 shows the NMSE performance of the algorithms under test after averaging the results of 100 Monte Carlo trials. Here, since $\mathbf{y}$ and $\mathbf{z} = \mathbf{A}\mathbf{x}$ are related through AWGN, the GAMP algorithm of [21] reduces to the Bayesian version of the AMP algorithm from [18].

Note that the case of i.i.d. $\mathbf{A}$ is the "ideal" scenario for both AMP and GAMP. As discussed in the Introduction, their convergence in this case is guaranteed rigorously through state evolution analysis [21]–[23] as $m, n \to \infty$. In Figure 2, since m and n are sufficiently large, it is not surprising to see that GAMP performs well over all measurement ratios m/n. Furthermore, it is interesting to notice that GAMP outperforms LASSO and obtains NMSEs that are very close to that of the support-aware genie. Under such ideal $\mathbf{A}$, the proposed ADMM-GAMP method matches the performance of GAMP (since it minimizes the same objective) but does not offer any additional benefit.

The benefits of ADMM-GAMP become apparent in our second experiment, which uses non-i.i.d. matrices $\mathbf{A}$. In describing the experiment, we first recall that [24] established that the convergence of GAMP can be predicted by the peak-to-average ratio of the squared singular values,

$$\kappa(\mathbf{A}) \triangleq \frac{\sigma_1^2(\mathbf{A})}{\sum_{i=1}^{r} \sigma_i^2(\mathbf{A})/r}, \tag{82}$$

where $r = \min\{m, n\}$ and $\sigma_i(\mathbf{A})$ is the i-th largest singular value of $\mathbf{A}$. When this ratio κ is sufficiently large, the algorithm will diverge. Thus, to test the robustness of ADMM-GAMP, we constructed a sequence of matrices $\mathbf{A}$ with varying κ, as follows. First, the left and right singular vectors of $\mathbf{A}$ were generated by drawing an $m \times n$ matrix with i.i.d. $\mathcal{N}(0, 1/m)$ entries and taking its singular-value decomposition. Then, the singular values of $\mathbf{A}$ were chosen by setting the largest at $\sigma_1(\mathbf{A}) = 1$ and logarithmically spacing each successive singular value to attain the desired peak-to-average ratio κ.

As a function of κ, the NMSE performance of the various algorithms under test is illustrated in Figure 3 for the case of $m = 600$ measurements. There it can be seen that, for larger values of κ, the NMSE performance of the original GAMP algorithm deteriorated, which was a result of the algorithm diverging. (Note that, in the plot, we capped the maximum NMSE to 0 dB for visual clarity.) The figure also shows that the SwAMP method achieved low NMSE over a wider range of κ ratios than the original GAMP method, but its performance also degraded for larger values of κ. The ADMM-GAMP method, however, converged over the entire range of κ values, achieving NMSE performance relatively close to the support-aware genie.

In our third and final experiment, we recover $\mathbf{x}$ from "one-bit" measurements $\mathbf{y} = \text{sgn}(\mathbf{A}\mathbf{x})$, where sgn is the *sign function*, as considered in, e.g., [38] and [51]. Here, we used $m = 2000$ measurements and generated the matrices $\mathbf{A}$ as in our second experiment. Figure 4 shows the NMSE performance of the various algorithms under test. The results in the figure illustrate that the original GAMP method diverged for $\kappa \geq 2$. However, both SwAMP and ADMM-GAMP recovered the solution for the whole range of κ without diverging, with ADMM-GAMP yielding slightly better NMSE (about 0.3 dB better) at higher values of κ.

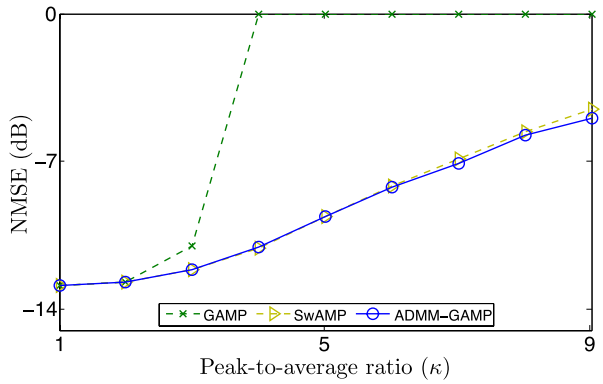

Fig. 4. Average NMSE versus peak-to-average squared-singular-value ratio $\kappa(\mathbf{A})$ when recovering a length $n = 1000$ Bernoulli-Gaussian signal $\mathbf{x}$ from $m = 2000$ noiseless 1-bit measurements $\mathbf{y} = \text{sgn}(\mathbf{Ax})$. Note the superior performance of ADMM-GAMP relative to the original GAMP and SwAMP.

X. Conclusions

Despite many promising results of AMP methods, the major stumbling block to more widespread use is their convergence and numerical stability. Although AMP techniques admit provable guarantees for i.i.d. $\mathbf{A}$, they can easily diverge for transforms that occur in many practical problems. While several methods have been proposed to improve the convergence, this paper provides a method with provable guarantees under arbitrary transforms. The method leverages well-established concepts of double-loop methods in belief propagation [31] as well as the classic ADMM method in optimization [9].

Nevertheless, there is still much work to be done. Most obviously, the proposed ADMM-GAMP method comes at a computational cost. Each iteration requires solving a (potentially large) least squares problem (50) that is not needed in the original AMP and GAMP algorithms. Similar to standard applications of ADMM, this minimization can likely be performed via conjugate gradient iterations, but its implementation requires further study. In any case, it is possible that ADMM-GAMP will be slower than other variants of GAMP. Indeed, our simulations suggest that other methods such as SwAMP or adaptively damped GAMP [27] may provide equally robust performance with less cost per iteration. One line of future work would thus be see to whether the proof techniques in this paper can be extended to address these algorithms as well.

The analysis in this paper might also be extended to other variants of AMP and GAMP. For example, it is conceivable that similar analysis could be applied to develop convergent approaches to the expectation-maximization (EM) GAMP developed in [39] and [52]–[54], turbo and hybrid GAMP methods in [56] and [57], and applications in dictionary learning and matrix factorization [57]–[59].

Appendix A
Proof of Theorem 1

Throughout this appendix, we use the shorthand notation for the gradient $h'(\boldsymbol{\tau}) \triangleq \partial h(\boldsymbol{\tau})/\partial \boldsymbol{\tau} \in \mathbb{R}^p$.

First we show, by induction, that $\boldsymbol{\gamma}^k \in \Gamma$ for all k. Recall that, by the hypothesis of the theorem, $\boldsymbol{\gamma}^0 \in \Gamma$. Now suppose that $\boldsymbol{\gamma}^k \in \Gamma$. Then the updates in Algorithm 1 imply that

$$h'(\boldsymbol{\tau}^k) = h'(\mathbf{g}(\mathbf{b}^k)) = h'(\mathbf{g}(\widehat{\mathbf{b}}(\boldsymbol{\gamma}^k))).$$

Then, by Assumption 1(c), $h'(\boldsymbol{\tau}^k) \in \Gamma$. Since $\boldsymbol{\gamma}^k \in \Gamma$, $\theta^k \in (0, 1]$, and Γ is convex,

$$\boldsymbol{\gamma}^{k+1} = (1-\theta^k)\boldsymbol{\gamma}^k + \theta^k h'(\boldsymbol{\tau}^k) \in \Gamma.$$

Thus, by induction, $\boldsymbol{\gamma}^k \in \Gamma$ for all k.

Next, we prove the decrementing property (27). First observe that since the restriction $\mathbf{b} \in B$ is a linear constraint, we can find a linear transform $\mathbf{B}$ and vector $\mathbf{b}_0$ such that $\mathbf{b} \in B$ if and only if $\mathbf{b} = \mathbf{Bx} + \mathbf{b}_0$ for some vector $\mathbf{x}$. It can be verified that we can reparameterize the functions $f(\cdot)$ and $\mathbf{g}(\cdot)$ around $\mathbf{x}$ and obtain the exact same recursions in Algorithm 1. Also, all the conditions in Assumptions 1 will hold for reparametrized functions as well. Thus, for the remainder of the proof we can ignore the linear constraints B, or alternatively view B as the entire vector space.

Under this assumption, for any $\boldsymbol{\gamma}$, and any minimizer $\widehat{\mathbf{b}}(\boldsymbol{\gamma})$ will be in the interior of B and therefore,

$$\mathbf{0} = \frac{\partial J(\widehat{\mathbf{b}}(\boldsymbol{\gamma}), \boldsymbol{\gamma})}{\partial \mathbf{b}} = f'(\widehat{\mathbf{b}}(\boldsymbol{\gamma})) + \sum_{\ell=1}^{L} g'_\ell(\widehat{\mathbf{b}}(\boldsymbol{\gamma}))\gamma_\ell \quad (83)$$

$$= f'(\widehat{\mathbf{b}}(\boldsymbol{\gamma})) + g'(\widehat{\mathbf{b}}(\boldsymbol{\gamma}))\boldsymbol{\gamma}, \quad (84)$$

where $f'(\mathbf{b})$ is shorthand notation for the gradient $\partial f(\mathbf{b})/\partial \mathbf{b}$, $g'_\ell(\mathbf{b})$ is shorthand for the gradient (with respect to $\mathbf{b}$) of the ℓth component of the vector-valued function $\mathbf{g}(\cdot)$, and where $g'(\mathbf{b}) = [g'_1(\mathbf{b}), \ldots, g'_L(\mathbf{b})]$ is matrix-valued. Taking the gradient of (83) with respect to $\boldsymbol{\gamma}^{\mathsf{T}}$ yields the matrix

$$\mathbf{0} \overset{(a)}{=} \frac{\partial f'(\widehat{\mathbf{b}}(\boldsymbol{\gamma}))}{\partial \boldsymbol{\gamma}^{\mathsf{T}}} + \sum_{\ell=1}^{L} \frac{\partial g'_\ell(\widehat{\mathbf{b}}(\boldsymbol{\gamma}))}{\partial \boldsymbol{\gamma}^{\mathsf{T}}}\gamma_\ell + g'(\widehat{\mathbf{b}}(\boldsymbol{\gamma}))$$

$$\overset{(b)}{=} \left[\frac{\partial f'(\widehat{\mathbf{b}}(\boldsymbol{\gamma}))}{\partial \widehat{\mathbf{b}}^{\mathsf{T}}} + \sum_{\ell=1}^{L} \frac{\partial g'_\ell(\widehat{\mathbf{b}}(\boldsymbol{\gamma}))}{\partial \widehat{\mathbf{b}}^{\mathsf{T}}}\gamma_\ell\right] \frac{\partial \widehat{\mathbf{b}}(\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}^{\mathsf{T}}} + g'(\widehat{\mathbf{b}}(\boldsymbol{\gamma}))$$

$$= \mathbf{H}(\boldsymbol{\gamma})\frac{\partial \widehat{\mathbf{b}}(\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}^{\mathsf{T}}} + g'(\widehat{\mathbf{b}}(\boldsymbol{\gamma})), \quad (85)$$

where (a) and (b) follow from the chain rule and $\mathbf{H}(\boldsymbol{\gamma})$ is the Hessian from (25). Equation (85) then implies

$$\frac{\partial \widehat{\mathbf{b}}(\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}^{\mathsf{T}}} = -\mathbf{H}(\boldsymbol{\gamma})^{-1} g'(\widehat{\mathbf{b}}(\boldsymbol{\gamma})), \quad (86)$$

where Assumption 1(b) guarantees the existence of the inverse. The gradient of the objective with respect to $\boldsymbol{\gamma}^{\mathsf{T}}$ is then

$$\frac{\partial J(\widehat{\mathbf{b}}(\boldsymbol{\gamma}))}{\partial \boldsymbol{\gamma}^{\mathsf{T}}} \overset{(a)}{=} \left[f'(\widehat{\mathbf{b}}(\boldsymbol{\gamma})) + g'(\widehat{\mathbf{b}}(\boldsymbol{\gamma}))h'(\boldsymbol{\tau})\right]^{\mathsf{T}} \frac{\partial \widehat{\mathbf{b}}(\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}^{\mathsf{T}}}$$

$$\overset{(b)}{=} \left(h'(\boldsymbol{\tau}) - \boldsymbol{\gamma}\right)^{\mathsf{T}} g'(\widehat{\mathbf{b}}(\boldsymbol{\gamma}))^{\mathsf{T}} \frac{\partial \widehat{\mathbf{b}}(\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}^{\mathsf{T}}}$$

$$\overset{(c)}{=} \left(\boldsymbol{\gamma} - h'(\boldsymbol{\tau})\right)^{\mathsf{T}} g'(\widehat{\mathbf{b}}(\boldsymbol{\gamma}))^{\mathsf{T}} \mathbf{H}(\boldsymbol{\gamma})^{-1} g'(\widehat{\mathbf{b}}(\boldsymbol{\gamma})), \quad (87)$$

where (a) follows from (22) and the chain rule, (b) follows from (84), and (c) follows from (86).

Notice that the $\boldsymbol{\gamma}^k$ update in Algorithm 1 can be written as

$$\boldsymbol{\gamma}^{k+1} - \boldsymbol{\gamma}^k = \big(h'(\boldsymbol{\tau}^k) - \boldsymbol{\gamma}^k\big)\theta^k.$$

Taking an inner product of the above and (87) evaluated at $\boldsymbol{\gamma} = \boldsymbol{\gamma}^k$, we get

$$\begin{aligned}
&\left[\frac{\partial J(\widehat{\mathbf{b}}(\boldsymbol{\gamma}^k))}{\partial \boldsymbol{\gamma}^{\mathsf{T}}}\right]\left(\boldsymbol{\gamma}^{k+1} - \boldsymbol{\gamma}^k\right)\\
&= -\big(\boldsymbol{\gamma}^k - h'(\boldsymbol{\tau}^k)\big)^{\mathsf{T}} g'(\mathbf{b}^k)^{\mathsf{T}}\mathbf{H}(\boldsymbol{\gamma}^k)^{-1} g'(\mathbf{b}^k)\big(\boldsymbol{\gamma}^k - h'(\boldsymbol{\tau}^k)\big)\theta^k\\
&\le -\frac{\theta^k}{c_2}\big\| g'(\mathbf{b}^k)\big(\boldsymbol{\gamma}^k - h'(\boldsymbol{\tau}^k)\big)\big\|^2, \qquad (88)
\end{aligned}$$

recalling that $\widehat{\mathbf{b}}(\gamma^k) = \mathbf{b}^k$ and that c_2 was defined in Assumption 1(b). Therefore, the update of $\boldsymbol{\gamma}^k$ is in a descent direction on the objective $J(\widehat{\mathbf{b}}(\boldsymbol{\gamma}))$. Hence, for a sufficiently small damping parameter θ^k, we will have

$$J(\mathbf{b}^{k+1}) - J(\mathbf{b}^k) = J(\widehat{\mathbf{b}}(\boldsymbol{\gamma}^{k+1})) - J(\widehat{\mathbf{b}}(\boldsymbol{\gamma}^k)) \le 0,$$

which proves the decrementing property (27).

Appendix B
Large Deviations View of MAP Estimation

For each $T > 0$, let $\mathbf{x}^t(T), \mathbf{z}^t(T), \ldots$, be the output of the ADMM-GAMP algorithm with the MMSE estimation functions (52) and the scaled penalties (68). Next, we define several limits. For the mean vectors we define

$$\mathbf{x}^t = \lim_{T\to 0} \mathbf{x}^t(T), \quad \mathbf{z}^t = \lim_{T\to 0} \mathbf{z}^t(T),$$

for the dual vectors we define

$$\mathbf{q}^t = \lim_{T\to 0} T\mathbf{q}^t(T), \quad \mathbf{s}^t = \lim_{T\to 0} T\mathbf{s}^t(T),$$

and for the variance terms we define

$$\boldsymbol{\tau}_x^t = \lim_{T\to 0} \frac{\boldsymbol{\tau}_x^t(T)}{T}, \boldsymbol{\tau}_z^t = \lim_{T\to 0} \frac{\boldsymbol{\tau}_z^t(T)}{T}, \boldsymbol{\tau}_p^t = \lim_{T\to 0} \frac{\boldsymbol{\tau}_p^t(T)}{T}, \qquad (89a)$$

$$\boldsymbol{\tau}_r^t = \lim_{T\to 0} \frac{\boldsymbol{\tau}_r^t(T)}{T}, \boldsymbol{\tau}_s^t = \lim_{T\to 0} T\boldsymbol{\tau}_s^t(T). \qquad (89b)$$

We will assume that all of these limits exist. Note that some of terms are scaled by T and others by $1/T$. These normalizations are important. It is easily checked that the scalings all cancel, so that the limiting values satisfy the recursions of Algorithm 3 with the limiting estimation functions

$$\begin{aligned}
g_x(\mathbf{r}, \boldsymbol{\tau}_r) &\triangleq \lim_{T\to 0} g_x(\mathbf{r}, \boldsymbol{\tau}_r(T); T) && (90a)\\
&= \lim_{T\to 0} g_x(\mathbf{r}, \boldsymbol{\tau}_r T; T), && (90b)\\
g_z(\mathbf{p}, \boldsymbol{\tau}_p) &\triangleq \lim_{T\to 0} g_z(\mathbf{p}, \boldsymbol{\tau}_p(T); T) && (90c)\\
&= \lim_{T\to 0} g_z(\mathbf{p}, \boldsymbol{\tau}_p T; T), && (90d)
\end{aligned}$$

where $g_x(\mathbf{r}, \boldsymbol{\tau}_r T; T)$ and $g_z(\mathbf{p}, \boldsymbol{\tau}_p T; T)$ are the MMSE estimation functions (52) for the scaled penalties (68). Note that we have used the scalings in (89), which show $\boldsymbol{\tau}_r(T) \approx T\boldsymbol{\tau}_r$ and $\boldsymbol{\tau}_p(T) \approx \boldsymbol{\tau}_p T$ for small T. Now, the scaled function $g_x(\mathbf{r}, \boldsymbol{\tau}_r T; T)$ is the expectation $\mathbb{E}(\mathbf{x}|T)$ with respect to the density

$$p(\mathbf{x}|\mathbf{r}, \boldsymbol{\tau}_r T; T) \propto \exp\left[-\frac{f_x(\mathbf{x})}{T} - \frac{1}{2T}\|\mathbf{x} - \mathbf{r}\|^2_{\boldsymbol{\tau}_r}\right].$$

Laplace's Principle [48] from large deviations theory shows that (under mild conditions) this density concentrates around its maxima, and thus the expectation with respect to this density converges to the minimum

$$\lim_{T\to 0} g_x(\mathbf{r}, \boldsymbol{\tau}_r T; T) = \arg\min_{\mathbf{x}} f_x(\mathbf{x}) + \tfrac{1}{2}\|\mathbf{x} - \mathbf{r}\|^2_{\boldsymbol{\tau}_r},$$

which is exactly the minimization in the MAP estimation function (63). The limit of $g_z(\mathbf{p}, \boldsymbol{\tau}_p/T; T)$ as $T \to 0$ is similar. We conclude that the limit of the ADMM-GAMP algorithm with MMSE estimation functions (52) and scaled densities (68) is exactly the ADMM-GAMP algorithm with the MAP estimation functions (63). In particular, for each T, the density over x_j in (48) is given by

$$b_{x_j}^t(x_j|r_j, \tau_{r_j} T) \propto \exp\left[-\frac{f_{x_j}(x_j)}{T} - \frac{(x_j - r_j^t)^2}{2T\tau_{r_j}^t}\right], \qquad (91)$$

from which we can prove the limits in (69).

It remains to show that the LSL-BFE in (16) with the scaled functions (68) decomposes into the optimizations (57) and (72) as $T \to 0$. To this end, let $J(b_x, b_z; T)$ be the LSL-BFE (16) for the scaled penalties (68), which is given by

$$\begin{aligned}
J(b_x, b_z; T) &= D(b_x \| Z_{x,T}^{-1} \mathrm{e}^{-f_x/T}) + D(b_z \| Z_{z,T}^{-1} \mathrm{e}^{-f_z/T})\\
&\quad + H(\mathrm{var}(\mathbf{x}|b_x), \mathrm{var}(\mathbf{z}|b_z))\\
&= \frac{1}{T}\Big[\mathbb{E}(f_x(\mathbf{x})|b_x) + \mathbb{E}(f_z(\mathbf{z})|b_z)\Big]\\
&\quad + H(\mathrm{var}(\mathbf{x}|b_x), \mathrm{var}(\mathbf{z}|b_z))\\
&\quad - H(b_x) - H(b_z) + \mathrm{const}, \qquad (92)
\end{aligned}$$

where $H(a)$ denotes the differential entropy of distribution a, $H(\boldsymbol{\tau}_x, \boldsymbol{\tau}_z)$ is the entropy bound from (17), $Z_{x,T} \triangleq \int \mathrm{e}^{-f_x(\mathbf{x})/T}\,\mathrm{d}\mathbf{x}$, $Z_{z,T} \triangleq \int \mathrm{e}^{-f_z(\mathbf{z})/T}\,\mathrm{d}\mathbf{z}$, and the "const" in (92) is with respect to b_x and b_z. Now, we know that, as $T \to 0$, the optimal densities b_x and b_z will concentrate around their maxima with variance $O(T)$. Thus, we can take a quadratic approximation around the maximum

$$\ln b_{x_j}(x_j) \approx \frac{(x_j - \widehat{x}_j)^2}{2T\tau_{x_j}} + \mathrm{const}, \qquad (93)$$

where

$$\widehat{x}_j = \arg\min_{x_j} -\ln b_{x_j}(x_j), \quad \frac{1}{\tau_{x_j}} = -T\frac{\partial^2 \ln b_{x_j}(x_j)}{\partial x_j^2},$$

with a similar approximation for $\ln b_{z_i}(z_i)$. Under these approximations, $b_{x_j}(x_j)$ and $b_{z_i}(z_i)$ become approximately Gaussian, i.e.,

$$b_{x_j}(x_j) \approx \mathcal{N}(\widehat{x}_j, T\tau_{x_j}), \quad b_{z_i}(z_i) \approx \mathcal{N}(\widehat{z}_i, T\tau_{z_i}). \qquad (94)$$

Using these Gaussian approximations, we can compute the expectations

$$\begin{aligned}
&\mathbb{E}(f_{x_j}(x_j)|b_{x_j}) \\
&\quad= \int f_{x_j}(x)\mathcal{N}(x;\widehat{x}_j, T\tau_{x_j})\,\mathrm{d}x \\
&\quad\overset{(a)}{=} \int \sum_{k=0}^{\infty} \frac{(x-\widehat{x}_j)^k f_{x_j}^{(k)}(\widehat{x}_j)}{k!}\mathcal{N}(x;\widehat{x}_j, T\tau_{x_j})\,\mathrm{d}x \quad (95)\\
&\quad\overset{(b)}{=} \sum_{k=0}^{\infty} \frac{f_{x_j}^{(k)}(\widehat{x}_j)}{k!}\int (x-\widehat{x}_j)^k\mathcal{N}(x;\widehat{x}_j, T\tau_{x_j})\,\mathrm{d}x \quad (96)\\
&\quad\overset{(c)}{=} \sum_{l=0}^{\infty} \frac{f_{x_j}^{(2l)}(\widehat{x}_j)}{(2l)!}(T\tau_{x_j})^l(2l-1)!! \quad (97)\\
&\quad\overset{(c)}{=} \sum_{l=0}^{\infty} \frac{f_{x_j}^{(2l)}(\widehat{x}_j)}{2^l l!}(T\tau_{x_j})^l, \quad (98)
\end{aligned}$$

where (a) wrote $f_{x_j}(x)$ using a Taylor series about $x = \widehat{x}_j$; (b) assumed the exchange of limit and integral; (c) used the expression for the Gaussian central moments, which involves the double factorial $(2l-1)!! = (2l-1)(2l-3)(2l-5)\times\cdots\times 1$; and (d) used the identity $(2l-1)!! = \frac{(2l)!}{2^l l!}$. Thus, for small T, we have

$$\mathbb{E}(f_{x_j}(x_j)|b_{x_j}) \approx f_{x_j}(\widehat{x}_j) + \frac{1}{2}T\tau_{x_j}f''_{x_j}(\widehat{x}_j), \quad (99a)$$

$$\mathbb{E}(f_{z_i}(z_i)|b_{z_i}) \approx f_{z_i}(\widehat{z}_i) + \frac{1}{2}T\tau_{z_i}f''_{z_i}(\widehat{z}_i). \quad (99b)$$

The differential entropies of these Gaussians (94) are

$$H(b_{x_j}) = \frac{1}{2}\ln(2\pi eT\tau_{x_j}), \quad H(b_{z_i}) = \frac{1}{2}\ln(2\pi eT\tau_{z_i}), \quad (100)$$

and the entropy term (17) is

$$\begin{aligned}
H(\mathrm{var}(\mathbf{x}|b_x), \mathrm{var}(\mathbf{z}|b_z)) &= H(T\boldsymbol{\tau}_x, T\boldsymbol{\tau}_z)\\
&= \frac{1}{2}\left[\sum_{i=1}^{m}\frac{\tau_{z_i}}{\tau_{p_i}} + \ln(2\pi T\tau_{p_i})\right]
\end{aligned} \quad (101)$$

for $\boldsymbol{\tau}_p \triangleq \mathbf{S}\boldsymbol{\tau}_x$. Substituting (99), (100) and (101) into (92), we obtain

$$J_T(b_x, b_z) = \frac{1}{T}J(\widehat{\mathbf{x}}, \widehat{\mathbf{z}}) + \frac{1}{2}J^2(\boldsymbol{\tau}_x, \boldsymbol{\tau}_z, \widehat{\mathbf{x}}, \widehat{\mathbf{z}}) + \mathrm{const}, \quad (102)$$

where $J(\cdot)$ and $J^2(\cdot)$ are given in (57) and (73). As $T \to 0$, the first term in (102) dominates, implying that the optimization of $(\widehat{\mathbf{x}}, \widehat{\mathbf{z}})$ can be conducted independently of $\boldsymbol{\tau}_x, \boldsymbol{\tau}_z$, as in (57). The subsequent optimization of $(\boldsymbol{\tau}_x, \boldsymbol{\tau}_z)$ then follows, as given in (72).

Appendix C
Proof of Theorems 2 and 3

We will just prove Theorem 2 since the proof of Theorem 3 is very similar. For the original constrained optimization (14), define the Lagrangian

$$L_0(b_x, b_z, \mathbf{s}) \triangleq J(b_x, b_z) + \mathbf{s}^{\mathsf{T}}(\mathbb{E}(\mathbf{z}|b_z) - \mathbf{A}\mathbb{E}(\mathbf{x}|b_x)). \quad (103)$$

We need to show that any fixed points $(b_x, b_z, \mathbf{s})$ of ADMM-GAMP are critical points of this Lagrangian.

First observe that, any fixed point, $\boldsymbol{\tau}_r$ from line 22 of Algorithm 3 satisfies

$$\mathbf{1}./(2\boldsymbol{\tau}_r) = \mathbf{1}./(2\overline{\boldsymbol{\tau}}_r) = \frac{\partial H(\boldsymbol{\tau}_x, \boldsymbol{\tau}_z)}{\partial \boldsymbol{\tau}_x}, \quad (104)$$

where the last step follows from the construction of $\overline{\boldsymbol{\tau}}_r$ in (32). Similarly, at any fixed point of line 23,

$$\mathbf{1}./(2\boldsymbol{\tau}_p) = \mathbf{1}./(2\overline{\boldsymbol{\tau}}_p) = \frac{\partial H(\boldsymbol{\tau}_x, \boldsymbol{\tau}_z)}{\partial \boldsymbol{\tau}_z}. \quad (105)$$

From (42b) and (42c), we see that any fixed point satisfies

$$\mathbb{E}(\mathbf{z}|b_z) = \mathbf{A}\mathbf{v}, \quad \mathbb{E}(\mathbf{x}|b_x) = \mathbf{v}. \quad (106)$$

Thus, the constraint in (14) is satisfied, in that $\mathbb{E}(\mathbf{z}|b_z) = \mathbf{A}\mathbb{E}(\mathbf{x}|b_x)$. Furthermore, since $\mathbf{v}$ minimizes (50), we know that it zeros the gradient of the corresponding cost function:

$$\begin{aligned}
\mathbf{0} = \mathbf{A}^{\mathsf{T}}\mathbf{D}_{\boldsymbol{\tau}_p}\big(\mathbb{E}(\mathbf{z}|b_z) - \mathbf{A}\mathbf{v} + \boldsymbol{\tau}_p.\mathbf{s}\big)\\
+\mathbf{D}_{\boldsymbol{\tau}_r}\big(\mathbb{E}(\mathbf{x}|b_x) - \mathbf{v} + \boldsymbol{\tau}_r.\mathbf{q}\big),
\end{aligned} \quad (107)$$

where $\mathbf{D}_{\boldsymbol{\tau}} = \mathrm{Diag}(\mathbf{1}./\boldsymbol{\tau})$. Plugging (106) into the previous expression, we obtain

$$\mathbf{q} = -\mathbf{A}^{\mathsf{T}}\mathbf{s}. \quad (108)$$

Since b_x minimizes the augmented Lagrangian in (42a), it zeros the corresponding gradient, i.e.,

$$\begin{aligned}
\mathbf{0} &= \frac{\partial}{\partial b_x}L(b_x, b_z, \mathbf{s}, \mathbf{q}, \mathbf{v}; \boldsymbol{\tau}_r, \boldsymbol{\tau}_p)\\
&\overset{(a)}{=} \frac{\partial}{\partial b_x}\Big[J(b_x, b_z, \boldsymbol{\tau}_r, \boldsymbol{\tau}_p) + \mathbf{q}^{\mathsf{T}}\mathbb{E}(\mathbf{x}|b_x)\\
&\qquad + \frac{1}{2}\|\mathbb{E}(\mathbf{x}|b_x) - \mathbf{v}\|^2_{\boldsymbol{\tau}_r}\Big]\\
&\overset{(b)}{=} \frac{\partial}{\partial b_x}\Big[J(b_x, b_z, \boldsymbol{\tau}_r, \boldsymbol{\tau}_p) - \mathbf{q}^{\mathsf{T}}\mathbb{E}(\mathbf{x}|b_x)\Big]\\
&\overset{(c)}{=} \frac{\partial}{\partial b_x}\Big[J(b_x, b_z, \boldsymbol{\tau}_r, \boldsymbol{\tau}_p) - \mathbf{s}^{\mathsf{T}}\mathbf{A}\mathbb{E}(\mathbf{x}|b_x)\Big]\\
&\overset{(d)}{=} \frac{\partial}{\partial b_x}\Big[J(b_x, b_z) - H\big(\mathrm{var}(\mathbf{x}|b_x), \boldsymbol{\tau}_z\big)\\
&\qquad + (\mathbf{1}./(2\boldsymbol{\tau}_r))^{\mathsf{T}}\mathrm{var}(\mathbf{x}|b_x) - \mathbf{s}^{\mathsf{T}}\mathbf{A}\mathbb{E}(\mathbf{x}|b_x)\Big]\\
&\overset{(e)}{=} \frac{\partial}{\partial b_x}\Big[J(b_x, b_z) - \mathbf{s}^{\mathsf{T}}\mathbf{A}\mathbb{E}(\mathbf{x}|b_x)\Big]\\
&\overset{(f)}{=} \frac{\partial}{\partial b_x}L_0(b_x, b_z, \mathbf{s}),
\end{aligned} \quad (109)$$

where (a) follows from substituting (41) and eliminating terms that do not depend on b_x, since their gradient equals zero; (b) follows from (106); (c) follows from (108); (d) follows from the definitions of the original and linearized LSL-BFEs in (16) and (31); (e) follows from the chain rule and the gradient in (104); and (f) follows from (103). A similar computation shows that

$$\frac{\partial}{\partial b_z}L_0(b_x, b_z, \mathbf{s}) = \mathbf{0}. \quad (110)$$

Together, (109) and (110) show that (b_x, b_z) are critical points of the Lagrangian $L_0(b_x, b_z, \mathbf{s})$ for the dual parameters $\mathbf{s}$. Since these densities also satisfy the constraint $\mathbb{E}(\mathbf{z}|b_z) = \mathbf{A}\mathbb{E}(\mathbf{x}|b_x)$, we conclude that (b_x, b_z) are critical points of the constrained optimization (14).

APPENDIX D
PROOF OF LEMMA 1

For the MAP estimation functions (63), we know that

$$\widehat{x}_j = g_{x_j}(r_j, \tau_{r_j}) = \arg\min_{x_j} f_{x_j}(x_j) + \frac{1}{2\tau_{r_j}}(x_j - r_j)^2,$$

which implies that $x_j = \widehat{x}_j$ is a solution to $0 = f'_{x_j}(x_j) + (x_j - r_j)/\tau_{r_j}$, i.e., that

$$\widehat{x}_j = r_j - \tau_{r_j} f'_{x_j}(\widehat{x}_j).$$

Taking the derivative with respect to r_j, we find

$$\frac{\partial \widehat{x}_j}{\partial r_j} = 1 - \tau_{r_j} f''_{x_j}(\widehat{x}_j)\frac{\partial \widehat{x}_j}{\partial r_j},$$

which can be rearranged to form

$$\frac{\partial \widehat{x}_j}{\partial r_j} = g'_{x_j}(r_j, \tau_{r_j}) = \frac{1}{1 + f''_{x_j}(\widehat{x}_j)\tau_{r_j}}. \tag{111}$$

Then, given the assumption in the lemma, (111) implies that

$$\frac{1}{1 + B\tau_{r_j}} \le g'_{x_j}(r_j, \tau_{r_j}) \le \frac{1}{1 + A\tau_{r_j}}.$$

A similar bound can be obtained for $g'_{z_i}(p_i, \tau_{p_i})$, which proves (74) for any fixed $\boldsymbol{\tau}_r$ and $\boldsymbol{\tau}_p$.

The proof for MMSE estimation functions (52) uses a classic result of log-concave functions [60]. Since the functions f_x and f_z are separable, so are the estimation functions g_x and g_z (52), as established in (53). In particular, we can write

$$g_{x_j}(r_j, \tau_{r_j}) = \mathbb{E}(x_j | r_j, \tau_{r_j}), \quad g_{z_i}(p_i, \tau_{p_i}) = \mathbb{E}(z_i | p_i, \tau_{p_i}),$$

where the expectations are with respect to the densities

$$p(x_j | r_j, \tau_{r_j}) \propto \exp\left[-f_{x_j}(x_j) - \frac{(x_j - r_j)^2}{2\tau_{r_j}}\right] \tag{112a}$$

$$p(z_i | p_i, \tau_{p_i}) \propto \exp\left[-f_{z_i}(z_i) - \frac{(z_i - p_i)^2}{2\tau_{p_i}}\right]. \tag{112b}$$

We then need to show that the condition (74) is satisfied for each of the functions g_{x_j} and g_{z_i}. Below, we prove this for g_{x_j}, noting that the proof for g_{z_i} is similar.

From (56), we know that the derivative of $g_{x_j}(r_j, \tau_{r_j})$ with respect to r_j is given by

$$g'_{x_j}(r_j, \tau_{r_j}) = \frac{\tau_{x_j}}{\tau_{r_j}}, \quad \tau_{x_j} = \mathrm{var}(x_j | r_j, \tau_{r_j}). \tag{113}$$

The variance here is with respect to the density (112a), which can be rewritten as

$$p(x_j | r_j, \tau_{r_j}) \propto \exp\big[-h(x_j)\big]$$

for the potential function

$$h(x_j) = f_{x_j}(x_j) + \frac{(x_j - r_j)^2}{2\tau_{r_j}},$$

which has second derivative

$$h''(x_j) = f''_{x_j}(x_j) + \frac{1}{\tau_{r_j}}.$$

By assumption (75), this derivative is bounded as

$$A + \frac{1}{\tau_{r_j}} \le h''(x_j) \le B + \frac{1}{\tau_{r_j}}.$$

In particular, $h(x_j)$ is strictly convex. From (113) and [60, Th. 4.1], we have that

$$g'_{x_j}(r_j, \tau_{r_j}) = \frac{\tau_{x_j}}{\tau_{r_j}} = \frac{\mathrm{var}(x_j | r_j, \tau_{r_j})}{\tau_{r_j}} \le \frac{1}{\tau_{r_j}}\mathbb{E}\left(\frac{1}{h''(x)}\right) \le \frac{1}{A\tau_{r_j} + 1}. \tag{114}$$

It is also shown in [61, eq. (4.13)] that

$$g'_{x_j}(r_j, \tau_{r_j}) = \frac{\tau_{x_j}}{\tau_{r_j}} = \frac{\mathrm{var}(x_j | r_j, \tau_{r_j})}{\tau_{r_j}} \ge \frac{1}{\mathbb{E}(h''(x_j))\tau_{r_j}} \ge \frac{1}{B\tau_{r_j} + 1}. \tag{115}$$

Thus, we conclude that

$$\frac{1}{1 + B\tau_{r_j}} \le g'_{x_j}(r_j, \tau_{r_j}) \le \frac{1}{1 + A\tau_{r_j}},$$

which proves (74).

APPENDIX E
PROOF OF THEOREM 4

We find it easier to analyze the algorithm after the variables are combined and scaled as

$$\boldsymbol{\tau} \triangleq \begin{bmatrix} \boldsymbol{\tau}_r \\ \boldsymbol{\tau}_p \end{bmatrix}, \quad \mathbf{D} \triangleq \mathrm{Diag}(\mathbf{1}./\boldsymbol{\tau}), \tag{116}$$

and

$$\mathbf{w} \triangleq \mathbf{D}^{1/2}\begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix}, \ \mathbf{u} \triangleq \mathbf{D}^{-1/2}\begin{bmatrix} \mathbf{q} \\ \mathbf{s} \end{bmatrix}, \ \mathbf{B} \triangleq \mathbf{D}^{1/2}\begin{bmatrix} \mathbf{I} \\ \mathbf{A} \end{bmatrix}. \tag{117}$$

Also, we define

$$g(\mathbf{w}, \boldsymbol{\tau}) \triangleq \begin{bmatrix} g_x(\mathbf{x}, \boldsymbol{\tau}_r) \\ g_z(\mathbf{z}, \boldsymbol{\tau}_p) \end{bmatrix}, \tag{118}$$

and henceforth suppress the dependence on $\boldsymbol{\tau}$ in the notation since $\boldsymbol{\tau}$ is constant in this analysis. The mean update steps in Algorithm 3 then become

$$\mathbf{w}^{t+1} = \mathbf{D}^{1/2} g(\mathbf{D}^{-1/2}(\mathbf{B}\mathbf{v}^t - \mathbf{u}^t)) \tag{119a}$$

$$\mathbf{u}^{t+1} = \mathbf{u}^t + \mathbf{w}^{t+1} - \mathbf{B}\mathbf{v}^t \tag{119b}$$

$$\mathbf{v}^{t+1} = \arg\min_{\mathbf{v}} \|\mathbf{w}^{t+1} + \mathbf{u}^{t+1} - \mathbf{B}\mathbf{v}\|^2, \tag{119c}$$

where the result of (119c) can be written explicitly as

$$\mathbf{v}^t = (\mathbf{B}^{\mathsf{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathsf{T}}(\mathbf{w}^t + \mathbf{u}^t). \tag{120}$$

Let us define

$$\mathbf{P} \triangleq \mathbf{B}(\mathbf{B}^{\mathsf{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathsf{T}}, \quad \mathbf{P}^{\perp} \triangleq \mathbf{I} - \mathbf{P}, \tag{121}$$

where $\mathbf{P}$ is an orthogonal projector operator onto the column space of $\mathbf{B}$ and $\mathbf{P}^{\perp}$ is the projection onto its orthogonal complement. Noting that $\mathbf{B}\mathbf{v}^t = \mathbf{P}(\mathbf{w}^t + \mathbf{u}^t)$, (119a) reduces to

$$\begin{aligned} \mathbf{w}^{t+1} &= \mathbf{D}^{1/2} g\left(\mathbf{D}^{-1/2}(\mathbf{P}(\mathbf{w}^t + \mathbf{u}^t) - \mathbf{u}^t)\right) \\ &= \mathbf{D}^{1/2} g\left(\mathbf{D}^{-1/2}(\mathbf{P}\mathbf{w}^t - \mathbf{P}^{\perp}\mathbf{u}^t)\right) \\ &= \tilde{g}(\mathbf{P}\mathbf{w}^t - \mathbf{P}^{\perp}\mathbf{u}^t), \end{aligned} \tag{122}$$

where

$$\tilde{g}(\mathbf{w}) \triangleq \mathbf{D}^{1/2} g(\mathbf{D}^{-1/2}\mathbf{w}). \tag{123}$$

Also, since $\mathbf{P}^{\perp}\mathbf{B} = \mathbf{0}$, (119b) implies that

$$\begin{aligned}\mathbf{P}^{\perp}\mathbf{u}^{t+1} &= \mathbf{P}^{\perp}\mathbf{u}^{t} + \mathbf{P}^{\perp}\mathbf{w}^{t+1} \\ &= \mathbf{P}^{\perp}\mathbf{u}^{t} + \mathbf{P}^{\perp}\tilde{g}(\mathbf{P}\mathbf{w}^{t} - \mathbf{P}^{\perp}\mathbf{u}^{t}).\end{aligned} \tag{124}$$

Now define the state vector

$$\boldsymbol{\theta}^{t} \triangleq \begin{bmatrix} \mathbf{P}\mathbf{w}^{t} \\ \mathbf{P}^{\perp}\mathbf{u}^{t} \end{bmatrix}. \tag{125}$$

Since $\mathbf{P}^2 = \mathbf{P}$ and $(\mathbf{P}^{\perp})^2 = \mathbf{P}^{\perp}$,

$$\begin{bmatrix} \mathbf{P} & -\mathbf{P}^{\perp} \end{bmatrix} \boldsymbol{\theta}^{t} = \mathbf{P}\mathbf{w}^{t} - \mathbf{P}^{\perp}\mathbf{u}^{t}.$$

Therefore, from (122) and (124), respectively, we have that

$$\mathbf{P}\mathbf{w}^{t+1} = \mathbf{P}\tilde{g}\big(\begin{bmatrix} \mathbf{P} & -\mathbf{P}^{\perp} \end{bmatrix} \boldsymbol{\theta}^{t}\big), \tag{126}$$

$$\mathbf{P}^{\perp}\mathbf{u}^{t+1} = \mathbf{P}^{\perp}\mathbf{u}^{t} + \mathbf{P}^{\perp}\tilde{g}\big(\begin{bmatrix} \mathbf{P} & -\mathbf{P}^{\perp} \end{bmatrix} \boldsymbol{\theta}^{t}\big). \tag{127}$$

From (125), (126), and (127), we see that the mean update steps in Algorithm 3 are characterized by the recursive system

$$\boldsymbol{\theta}^{t+1} = f(\boldsymbol{\theta}^{t}) \tag{128}$$

for

$$f(\boldsymbol{\theta}) \triangleq \begin{bmatrix} \mathbf{P} \\ \mathbf{P}^{\perp} \end{bmatrix} \tilde{g}\big(\begin{bmatrix} \mathbf{P} & -\mathbf{P}^{\perp} \end{bmatrix} \boldsymbol{\theta}\big) + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^{\perp} \end{bmatrix} \boldsymbol{\theta}. \tag{129}$$

The following is a standard contraction mapping result [61]: if f has a continuous Jacobian f' whose spectral norm is less than one, i.e., $\exists \epsilon > 0$ s.t. $\|f'(\boldsymbol{\theta})\| < 1 - \epsilon \ \forall \boldsymbol{\theta}$, then the system (128) converges to a unique fixed point, $\boldsymbol{\theta}^{*}$, with a *linear* convergence rate, i.e.,

$$\exists C > 0 \text{ s.t. } \|\boldsymbol{\theta}^{t} - \boldsymbol{\theta}^{*}\| \leq C(1-\epsilon)^{t}.$$

So, our proof will be complete if we can show that the Jacobian of f from (129) is indeed a contraction.

First observe that, from the definition of $g(\mathbf{w})$ in (118), and the separability and boundedness assumptions in Assumption 2, the Jacobian of $g(\mathbf{w})$ at any $\mathbf{w}$ is diagonal and bounded:

$$\exists \epsilon \in (0, 0.5] \text{ s.t. } g'(\mathbf{w}) = \mathrm{Diag}(\mathbf{d}) \text{ and } \epsilon \leq d_k \leq 1 - \epsilon \ \forall k.$$

Since $\mathbf{D} = \mathrm{Diag}(\mathbf{1}./\boldsymbol{\tau})$ is also diagonal, the Jacobian of $\tilde{g}(\mathbf{w})$ in (123) is given by

$$\tilde{g}'(\mathbf{w}) = \mathbf{D}^{-1/2}\,\mathrm{Diag}(\mathbf{d})\mathbf{D}^{1/2} = \mathrm{Diag}(\mathbf{d}),$$

and hence

$$\epsilon\mathbf{I} \leq \tilde{g}'(\mathbf{w}) \leq (1-\epsilon)\mathbf{I} \tag{130}$$

for all $\mathbf{w}$. Now, the Jacobian of $f(\boldsymbol{\theta})$ in (129) is given by

$$f'(\boldsymbol{\theta}) = \begin{bmatrix} \mathbf{P} \\ \mathbf{P}^{\perp} \end{bmatrix} \tilde{g}'(\mathbf{w}) \begin{bmatrix} \mathbf{P} & -\mathbf{P}^{\perp} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^{\perp} \end{bmatrix}. \tag{131}$$

Hence, if we define

$$\mathbf{J}(\boldsymbol{\theta}) \triangleq f'(\boldsymbol{\theta}) \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} \end{bmatrix}, \tag{132}$$

then $\|f'(\boldsymbol{\theta})\| = \|\mathbf{J}(\boldsymbol{\theta})\|$ so $f'(\boldsymbol{\theta})$ is a contraction if and only if $\mathbf{J}(\boldsymbol{\theta})$ is. Therefore, it suffices to prove that $\mathbf{J}(\boldsymbol{\theta})$ is a contraction. Combining (131) and (132), we obtain

$$\mathbf{J}(\boldsymbol{\theta}) = \mathbf{U}^{\mathsf{T}}\tilde{g}'(\mathbf{w})\mathbf{U} - \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^{\perp} \end{bmatrix}, \tag{133}$$

where $\mathbf{w} = \begin{bmatrix} \mathbf{P} & \mathbf{P}^{\perp} \end{bmatrix} \boldsymbol{\theta}$, and

$$\mathbf{U} = \begin{bmatrix} \mathbf{P} & \mathbf{P}^{\perp} \end{bmatrix}. \tag{134}$$

Since $\mathbf{P}$ is an orthogonal projection and $\mathbf{P}^{\perp}$ is the projection onto the orthogonal complement, $\mathbf{U}$ is an isometry. That is,

$$\mathbf{U}\mathbf{U}^{\mathsf{T}} = \mathbf{P} + \mathbf{P}^{\perp} = \mathbf{I}, \tag{135}$$

and hence $\mathbf{U}^{\mathsf{T}}\mathbf{U} \leq \mathbf{I}$. Therefore, from (133) and (130),

$$\mathbf{J}(\boldsymbol{\theta}) \leq \mathbf{U}^{\mathsf{T}}\tilde{g}'(\mathbf{w})\mathbf{U} \leq (1-\epsilon)\mathbf{U}^{\mathsf{T}}\mathbf{U} \leq (1-\epsilon)\mathbf{I}. \tag{136}$$

For the lower bound, observe that

$$\begin{aligned}\mathbf{J}(\boldsymbol{\theta}) &\overset{(a)}{=} \mathbf{U}^{\mathsf{T}}\tilde{g}'(\mathbf{w})\mathbf{U} - \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^{\perp} \end{bmatrix} \\ &\overset{(b)}{\geq} \epsilon\mathbf{U}^{\mathsf{T}}\mathbf{U} - \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}^{\perp} \end{bmatrix} \\ &\overset{(c)}{=} \begin{bmatrix} \epsilon\mathbf{P} & \mathbf{0} \\ \mathbf{0} & (\epsilon-1)\mathbf{P}^{\perp} \end{bmatrix} \\ &\overset{(d)}{=} \begin{bmatrix} (\epsilon-1)\mathbf{I} & \mathbf{0} \\ \mathbf{0} & (\epsilon-1)\mathbf{I} \end{bmatrix} + \begin{bmatrix} \mathbf{I} - \epsilon\mathbf{P}^{\perp} & \mathbf{0} \\ \mathbf{0} & (1-\epsilon)\mathbf{P} \end{bmatrix} \\ &\geq (\epsilon-1)\mathbf{I},\end{aligned} \tag{137}$$

where step (a) follows from (133); (b) follows from (130); (c) follows from the definition of $\mathbf{U}$ in (134) and the fact that $\mathbf{P}$ and $\mathbf{P}^{\perp}$ are orthogonal projections; (d) follows from the definition of $\mathbf{P}^{\perp}$ in (121); and (137) follows because the eigenvalues of $\mathbf{P}^{\perp}$ and $\mathbf{P}$ are in the interval $[0, 1]$ and because $\epsilon \in (0, 0.5]$. Together (136) and (137) show that

$$\|f'(\boldsymbol{\theta})\| = \|\mathbf{J}(\boldsymbol{\theta})\| \leq 1 - \epsilon.$$

Hence the $f'(\boldsymbol{\theta})$ is a contraction and the ADMM-GAMP algorithm converges linearly at rate $1 - \epsilon$.

Appendix F
Proof of Theorem 5

We need to prove that the conditions of Assumption 1 are satisfied. Property (a) is satisfied since Theorem 4 shows that the constrained linearized LSL-BFE optimization (37) has a unique minima for any $(\boldsymbol{\tau}_r, \boldsymbol{\tau}_p) > 0$.

We next construct the set Γ. From the proof of Lemma 1, we know that when $\boldsymbol{\tau}_x = \mathrm{var}(\mathbf{x}|\mathbf{r}, \boldsymbol{\tau}_r)$ and $\boldsymbol{\tau}_z = \mathrm{var}(\mathbf{z}|\mathbf{p}, \boldsymbol{\tau}_p)$,

$$\boldsymbol{\tau}_x \in \left[\frac{A\boldsymbol{\tau}_r}{A+\boldsymbol{\tau}_r}, \frac{B\boldsymbol{\tau}_r}{B+\boldsymbol{\tau}_r}\right], \tag{138}$$

$$\boldsymbol{\tau}_z \in \left[\frac{A\boldsymbol{\tau}_p}{A+\boldsymbol{\tau}_p}, \frac{B\boldsymbol{\tau}_p}{B+\boldsymbol{\tau}_p}\right], \tag{139}$$

Hence

$$\boldsymbol{\tau}_s \triangleq \left(1 - \frac{\boldsymbol{\tau}_z}{\boldsymbol{\tau}_p}\right)\frac{1}{\boldsymbol{\tau}_p} \in \left[\frac{1}{B+\boldsymbol{\tau}_p}, \frac{1}{A+\boldsymbol{\tau}_p}\right]. \tag{140}$$

Now consider a set Γ of the form

$$\Gamma = \left\{(\boldsymbol{\tau}_r, \boldsymbol{\tau}_p) \;\middle|\; \boldsymbol{\tau}_r \in [a_r, b_r], \quad \boldsymbol{\tau}_p \in [a_p, b_p]\right\}. \tag{141}$$

In order that Γ satisfies Assumption 1(c), we need to find bounds a_r, b_r, a_p, b_p, such that if $(\boldsymbol{\tau}_r, \boldsymbol{\tau}_p) \in \Gamma$, then $(\overline{\boldsymbol{\tau}}_r, \overline{\boldsymbol{\tau}}_p) \in \Gamma$ where $(\overline{\boldsymbol{\tau}}_r, \overline{\boldsymbol{\tau}}_p)$ are given in (36).

To this end, first observe that (138) shows that $\boldsymbol{\tau}_x \leq 1/B$, so $\overline{\boldsymbol{\tau}}_p = \mathbf{S}\boldsymbol{\tau}_x \leq b_p$ for some b_p. If $\boldsymbol{\tau}_p \leq b_p$, (140) shows that $\boldsymbol{\tau}_s \in [1/B, 1/(A + b_p)]$. Therefore, using the boundedness assumptions on $\mathbf{S}$, $\overline{\boldsymbol{\tau}}_r = \mathbf{1}./(\mathbf{S}^{\mathsf{T}}\boldsymbol{\tau}_s) \in [a_r, b_r]$ for some lower and upper bounds a_r and b_r. Finally, if $\boldsymbol{\tau}_r \geq a_r$, $\boldsymbol{\tau}_x \geq Aa_r/(A + a_r)$ and hence $\overline{\boldsymbol{\tau}}_p = \mathbf{S}\boldsymbol{\tau}_x \geq a_p$ for some a_p. We conclude that we can find bounds a_r, b_r, a_p, b_p, such that if $(\boldsymbol{\tau}_r, \boldsymbol{\tau}_p) \in \Gamma$, then $(\overline{\boldsymbol{\tau}}_r, \overline{\boldsymbol{\tau}}_p) \in \Gamma$, and Γ is a compact, convex set satisfying Assumption 1(c).

Finally, we need to show the convexity assumption in Assumption 1(b). The linearized LSL-BFE in (31) is separable, so we only need to consider the convexity of one of the terms. To this consider a prototypical term of the form

$$J(b) = D(b\|\mathrm{e}^{-f_x}) + \frac{1}{2\tau_r}\mathrm{var}(x|b), \tag{142}$$

where $b(x)$ is some density over a scalar variable x and $f_x(x)$ is a convex penalty function. The Hessian of $J(b)$ is a quadratic form that takes perturbations $v_1(x)$ and $v_2(x)$ to the density $b(x)$ and returns a scalar. We will denote this Hessian by $J''(b)(v_1, v_2)$. Differentiating (142) we obtain that

$$J''(b)(v, v) = \int \frac{v(x)^2}{b(x)}\,\mathrm{d}x - \frac{1}{\tau_r}\left(\int v(x)x\,\mathrm{d}x\right)^2. \tag{143}$$

We need to show that this is positive. For any $v(x)$, let $u(x) = v(x)/b(x)$ so that $v(x) = u(x)b(x)$. Since a perturbation to the density $b(x)$ must satisfy $\int v(x)\,\mathrm{d}x = 0$, we have that

$$\mathbb{E}(u(x)|b) = \int u(x)b(x)\,\mathrm{d}x = 0.$$

Also, $J''(b)(v, v)$ above can be written as

$$\begin{aligned} J''(b)(v, v) &\overset{(a)}{=} \mathbb{E}(u(x)^2|b) - \frac{1}{\tau_r}\mathbb{E}(u(x)x|b) \\ &\overset{(b)}{=} \mathrm{var}(u(x)|b) - \frac{1}{\tau_r}\mathbb{E}^2(u(x)(x - \mu_x)|b) \\ &\overset{(c)}{\geq} \mathrm{var}(u(x)|b)\left[1 - \frac{\tau_x}{\tau_r}\right], \end{aligned} \tag{144}$$

where (a) follows from substituting $v(x) = u(x)b(x)$ into (143); in (b) we have used the notation $\mu_x = \mathbb{E}(x|b)$ and the fact that $\mathbb{E}(u(x)|b) = 0$; and (c) follows from the Cauchy-Schwartz inequality with the notation $\tau_x = \mathrm{var}(x|b)$. Now, using (138), we see that when $(\boldsymbol{\tau}_r, \boldsymbol{\tau}_p) \in \Gamma$, we have the lower bound,

$$1 - \frac{\tau_x}{\tau_r} \geq 1 - \frac{B}{B + a_r} \geq \frac{a_r}{B + a_r} > 0.$$

We conclude that there exists an ϵ such that

$$J''(b) \geq \epsilon\mathbf{I},$$

at any minima $b = \widehat{b}$ to the linearized LSL-BFE when $(\boldsymbol{\tau}_r, \boldsymbol{\tau}_p) \in \Gamma$. This proves Assumption 1(b). The uniform boundedness of all the other derivatives follows from the fact that all the terms are twice differentiable and the set Γ is compact.

Thus, all the conditions of Assumption 1 and the theorem follows from Theorem 1.

Appendix G
Proof of Theorem 6

We begin with proving part (a). We use induction. Suppose that (78) is satisfied for some t. Since $\mathbf{q}^0$, $\mathbf{x}^0$, and $\mathbf{v}^0$ are fixed points, we have from line 10 of Algorithm 3 that $\mathbf{x}^0 = \mathbf{v}^0$. Then, since $\mathbf{x}^0$ is a fixed point, we have from lines 7 and 9 and equation (63) that

$$\mathbf{x}^0 = g_x(\mathbf{r}^0, \boldsymbol{\tau}_r^0) = \arg\min_{\mathbf{x}} f_x(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \mathbf{v}^0 + \boldsymbol{\tau}_r^0.\mathbf{q}^0\|^2_{\boldsymbol{\tau}_r^0}.$$

Therefore, $\mathbf{x} = \mathbf{x}^0$ is the unique solution to

$$\mathbf{0} = f_x'(\mathbf{x}) + \mathrm{Diag}(\mathbf{1}./(2\boldsymbol{\tau}_r^0))(\mathbf{x} - \mathbf{x}^0) + \mathbf{q}^0,$$

which implies

$$f_x'(\mathbf{x}^0) = -\mathbf{q}^0. \tag{145}$$

By the induction hypothesis (78), $\mathbf{x}^t = \mathbf{x}^0$ and $\mathbf{q}^t = \mathbf{q}^0$. Since $\mathbf{x}^{t+1} = g_x(\mathbf{r}^t, \boldsymbol{\tau}_r^t)$, we have $\mathbf{x} = \mathbf{x}^{t+1}$ is the unique solution to

$$\begin{aligned} \mathbf{0} &= f'(\mathbf{x}) + \mathrm{Diag}(\mathbf{1}./(2\boldsymbol{\tau}_r^t))(\mathbf{x} - \mathbf{r}^t) && (146)\\ &= f'(\mathbf{x}) + \mathrm{Diag}(\mathbf{1}./(2\boldsymbol{\tau}_r^t))(\mathbf{x} - \mathbf{x}^0) + \mathbf{q}^0, && (147) \end{aligned}$$

where we have used the fact that

$$\mathbf{r}^t = \mathbf{x}^t + \boldsymbol{\tau}_r^t.\mathbf{q}^t = \mathbf{x}^0 + \boldsymbol{\tau}_r^t.\mathbf{q}^0.$$

From (145), $\mathbf{x} = \mathbf{x}^0$ is also a solution to (146). Therefore, $\mathbf{x}^{t+1} = \mathbf{x}^0$. Similarly, if $\mathbf{s}^t = \mathbf{s}^0$ and $\mathbf{z}^t = \mathbf{z}^0$, then $\mathbf{z}^{t+1} = \mathbf{z}^0$. From (50), $\mathbf{v}^{t+1} = \mathbf{v}^0$. We conclude that if (78) is satisfied for some t, it is satisfied for t+1. So part (a) follows by induction.

To prove part (b), we leverage the convergence result from [62]. Using our earlier result (111), we have that

$$\tau_{x_j}^{t+1} = \tau_{r_j}^t g_{x_j}'(r_j^t, \tau_{r_j}^t) = \frac{\tau_{r_j}^t}{1 + f_{x_j}''(x_j^{t+1})\tau_{r_j}^t}.$$

Rewriting this in vector form and using the updates in Algorithm 3 with $\theta^t = 1$, we obtain that

$$\begin{aligned} \mathbf{1}./\boldsymbol{\tau}_x^{t+1} &= \mathbf{1}./\boldsymbol{\tau}_r^t + f_x''(\mathbf{x}^{t+1}) = \mathbf{S}^{\mathsf{T}}\boldsymbol{\tau}_s^t + f_x''(\mathbf{x}^{t+1}) \\ &= \mathbf{S}^{\mathsf{T}}\boldsymbol{\tau}_s^t + \boldsymbol{\xi}_x, \quad \boldsymbol{\xi}_x \triangleq f_x''(\mathbf{x}^{t+1}) > \mathbf{0} \end{aligned} \tag{148}$$

where $f_x''(\mathbf{x}) = [f_{x_1}''(x_1), \ldots, f_{x_n}''(x_n)]^{\mathsf{T}}$ and where $\boldsymbol{\xi}_x$ is positive due to the convexity assumption and invariant to t due to part (a). Similarly, for the output estimation function g_z,

$$\boldsymbol{\tau}_z^{t+1} = \boldsymbol{\tau}_p^t.g_z'(\mathbf{p}^t, \boldsymbol{\tau}_p^t) = \boldsymbol{\tau}_p^t./(\mathbf{1} + f_z''(\mathbf{z}^{t+1}).\boldsymbol{\tau}_p^t).$$

Therefore, from the modified update of $\boldsymbol{\tau}_s^{t+1}$ in (77),

$$\boldsymbol{\tau}_s^{t+1} = f_z''(\mathbf{z}^{t+1})./(\mathbf{1} + f_z''(\mathbf{z}^{t+1}).\boldsymbol{\tau}_p^t),$$

or equivalently,

$$\begin{aligned} \mathbf{1}./\boldsymbol{\tau}_s^{t+1} &= \boldsymbol{\tau}_p^t + \mathbf{1}./f_z''(\mathbf{z}^{t+1}) = \mathbf{S}\boldsymbol{\tau}_x^t + \boldsymbol{\xi}_z && (149)\\ \boldsymbol{\xi}_z &\triangleq \mathbf{1}./f_z''(\mathbf{z}^{t+1}). && (150) \end{aligned}$$

Now define the maps,

$$\Phi_s(\boldsymbol{\tau}_x) := \mathbf{1}./\big(\mathbf{S}\boldsymbol{\tau}_x + \boldsymbol{\xi}_z\big)$$
$$\Phi_x(\boldsymbol{\tau}_s) := \mathbf{1}./\big(\mathbf{S}^{\mathsf{T}}\boldsymbol{\tau}_s + \boldsymbol{\xi}_x\big)$$

so that the updates (149) and (148) can be written as

$$\boldsymbol{\tau}_s^t = \Phi_s(\boldsymbol{\tau}_x^{t-1}), \quad \boldsymbol{\tau}_x^{t+1} = \Phi_x(\boldsymbol{\tau}_s^t).$$

Note that, due to part (a), $\boldsymbol{\xi}_x$ and $\boldsymbol{\xi}_z$ in (148) and (149) do not change with t. It is easy to check that, for any $\mathbf{S} > 0$,

(i) $\Phi_s(\boldsymbol{\tau}_x) > 0$,
(ii) $\boldsymbol{\tau}_x \geq \boldsymbol{\tau}_x' \Rightarrow \Phi_s(\boldsymbol{\tau}_x) \leq \Phi_s(\boldsymbol{\tau}_x')$, and
(iii) For all $\alpha > 1$, $\Phi_s(\alpha\boldsymbol{\tau}_x) > (1/\alpha)\Phi_s(\boldsymbol{\tau}_x)$.

with the analogous properties being satisfied by $\Phi_x(\boldsymbol{\tau}_s)$. Now let $\Phi := \Phi_x \circ \Phi_s$ be the composition of the two functions so that $\boldsymbol{\tau}_x^{t+1} = \Phi(\boldsymbol{\tau}_x^{t-1})$. Then, Φ satisfies the three properties:

(i) $\Phi(\boldsymbol{\tau}_x) > 0$,
(ii) $\boldsymbol{\tau}_x \geq \boldsymbol{\tau}_x' \Rightarrow \Phi(\boldsymbol{\tau}_x) \geq \Phi(\boldsymbol{\tau}_x')$, and
(iii) For all $\alpha > 1$, $\Phi(\alpha\boldsymbol{\tau}_x) < \alpha\Phi(\boldsymbol{\tau}_x)$.

Also, for any $\boldsymbol{\tau}_s \geq 0$, we have $\Phi_x(\boldsymbol{\tau}_s) \leq \mathbf{1}./\boldsymbol{\xi}_x$, and therefore, $\Phi(\boldsymbol{\tau}_x) \leq \mathbf{1}./\boldsymbol{\xi}_x$ for all $\boldsymbol{\tau}_x \geq 0$. Hence, taking any $\boldsymbol{\tau}_x \geq \mathbf{1}./\boldsymbol{\xi}_x$, we obtain:

$$\boldsymbol{\tau}_x \geq \Phi(\boldsymbol{\tau}_x).$$

The results in [62] then show that the updates $\boldsymbol{\tau}_x^{t+1} = \Phi(\boldsymbol{\tau}_x^{t-1})$ converge to unique fixed points. Since the increment increases by two, we need to apply the convergence twice: once for the $\boldsymbol{\tau}_x^t$ with odd values of t, and a second time for even values. Since the limit points are unique, both the even and odd subsequences will converge to the same value. A similar argument shows that $\boldsymbol{\tau}_s^t$ also converges to unique fixed points.

APPENDIX H
ORIGINAL GAMP VIA STALE, LINEARIZED ADMM

First, we examine the minimization in (80b). Starting with (79), a derivation identical to (44), but with $\mathbf{x}^{t+1} = \mathbb{E}(\mathbf{x}|b_x^{t+1})$ in place of $\mathbf{v}$, yields

$$\begin{aligned}
&L(b_x^{t+1}, b_z, \mathbf{s}^t; \boldsymbol{\tau}_p)\\
&= J(b_x^{t+1}, b_z, \boldsymbol{\tau}_r, \boldsymbol{\tau}_p) + (\mathbf{s}^t)^{\mathsf{T}}(\mathbb{E}(\mathbf{z}|b_z) - \mathbf{A}\mathbf{x}^{t+1})\\
&\quad + \tfrac{1}{2}\big\|\mathbb{E}(\mathbf{z}|b_z) - \mathbf{A}\mathbf{x}^{t+1}\big\|_{\boldsymbol{\tau}_p}^2 \qquad (151)\\
&= D(b_z\|Z_z^{-1}\mathrm{e}^{-f_z}) + \big(\mathbf{1}./(2\boldsymbol{\tau}_p)\big)^{\mathsf{T}}\mathrm{var}(\mathbf{z}|b_z) \qquad (152)\\
&\quad + \mathbb{E}\big(\tfrac{1}{2}\|\mathbf{z} - (\mathbf{A}\mathbf{x}^{t+1} - \boldsymbol{\tau}_p.\mathbf{s}^t)\|_{\boldsymbol{\tau}_p}^2\big|b_z\big) - \sum_{i=1}^m \frac{\tau_{z_i}}{2\tau_{p_i}} + \mathrm{const},\\
&= D(b_z\|Z_z^{-1}\mathrm{e}^{-f_z}) + \tfrac{1}{2}\mathbb{E}\big(\|\mathbf{z} - (\mathbf{A}\mathbf{x}^{t+1} - \boldsymbol{\tau}_p.\mathbf{s}^t)\|_{\boldsymbol{\tau}_p}^2\big|b_z\big)\\
&\quad + \mathrm{const},\\
&= \int_{\mathbb{R}^m} b_z(\mathbf{z}) \ln \frac{b_z(\mathbf{z})}{\exp(-f_z(\mathbf{z}) - \frac{1}{2}\|\mathbf{z} - (\mathbf{A}\mathbf{x}^{t+1} - \boldsymbol{\tau}_p.\mathbf{s}^t)\|_{\boldsymbol{\tau}_p}^2)}\,\mathrm{d}\mathbf{z}\\
&\quad + \mathrm{const} \qquad (153)\\
&= D(b_z\|p_z) + \mathrm{const}, \qquad (154)
\end{aligned}$$

where "const" is constant with respect to b_z and $p_z(\mathbf{z}) \propto \exp(-f_z(\mathbf{z}) - \frac{1}{2}\|\mathbf{z} - (\mathbf{A}\mathbf{x}^{t+1} - \boldsymbol{\tau}_p.\mathbf{s}^t)\|_{\boldsymbol{\tau}_p}^2)$. Thus, the minimizing density b_z output by (80b) is

$$b_z^{t+1}(\mathbf{z}) \propto \exp\big(-f_z(\mathbf{z}) - \tfrac{1}{2}\|\mathbf{z} - \mathbf{p}^{t+1}\|_{\boldsymbol{\tau}_p}^2\big) \qquad (155)$$
$$\mathbf{p}^{t+1} \triangleq \mathbf{A}\mathbf{x}^{t+1} - \boldsymbol{\tau}_p.\mathbf{s}^t. \qquad (156)$$

Next we examine the minimization in (80a). The objective function in (80a) can be written, using (79), $\mathbf{x}^t = \mathbb{E}(\mathbf{x}|b_x^t)$, and (31), as follows:

$$\begin{aligned}
&L(b_x, b_z^t, \mathbf{s}^{t-1}; \boldsymbol{\tau}_p)\\
&\quad + \tfrac{1}{2}\big(\mathbb{E}(\mathbf{x}|b_x) - \mathbf{x}^t\big)^{\mathsf{T}}\big(\mathbf{D}_{\boldsymbol{\tau}_r} - \mathbf{A}^{\mathsf{T}}\mathbf{D}_{\boldsymbol{\tau}_p}\mathbf{A}\big)\big(\mathbb{E}(\mathbf{x}|b_x) - \mathbf{x}^t\big)\\
&= D(b_x\|\mathrm{e}^{-f_x}) + \big(\mathbf{1}./\boldsymbol{\tau}_r\big)^{\mathsf{T}}\mathrm{var}(\mathbf{x}|b_x) - (\mathbf{s}^{t-1})^{\mathsf{T}}\mathbf{A}\mathbb{E}(\mathbf{x}|b_x)\\
&\quad + \tfrac{1}{2}\mathbb{E}(\mathbf{x}|b_x)^{\mathsf{T}}\mathbf{A}^{\mathsf{T}}\mathbf{D}_{\boldsymbol{\tau}_p}\mathbf{A}\mathbb{E}(\mathbf{x}|b_x) - (\mathbf{z}^t)^{\mathsf{T}}\mathbf{D}_{\boldsymbol{\tau}_p}\mathbf{A}\mathbb{E}(\mathbf{x}|b_x)\\
&\quad + \tfrac{1}{2}\mathbb{E}(\mathbf{x}|b_x)^{\mathsf{T}}\big(\mathbf{D}_{\boldsymbol{\tau}_r} - \mathbf{A}^{\mathsf{T}}\mathbf{D}_{\boldsymbol{\tau}_p}\mathbf{A}\big)\mathbb{E}(\mathbf{x}|b_x)\\
&\quad - (\mathbf{x}^t)^{\mathsf{T}}\big(\mathbf{D}_{\boldsymbol{\tau}_r} - \mathbf{A}^{\mathsf{T}}\mathbf{D}_{\boldsymbol{\tau}_p}\mathbf{A}\big)\mathbb{E}(\mathbf{x}|b_x) + \mathrm{const}\\
&= D(b_x\|\mathrm{e}^{-f_x}) + \big(\mathbf{1}./\boldsymbol{\tau}_r\big)^{\mathsf{T}}\mathrm{var}(\mathbf{x}|b_x) - (\mathbf{s}^{t-1})^{\mathsf{T}}\mathbf{A}\mathbb{E}(\mathbf{x}|b_x)\\
&\quad - (\mathbf{z}^t)^{\mathsf{T}}\mathbf{D}_{\boldsymbol{\tau}_p}\mathbf{A}\mathbb{E}(\mathbf{x}|b_x) + \tfrac{1}{2}\mathbb{E}(\mathbf{x}|b_x)^{\mathsf{T}}\mathbf{D}_{\boldsymbol{\tau}_r}\mathbb{E}(\mathbf{x}|b_x)\\
&\quad - (\mathbf{x}^t)^{\mathsf{T}}\mathbf{D}_{\boldsymbol{\tau}_r}\mathbb{E}(\mathbf{x}|b_x) + (\mathbf{x}^t)^{\mathsf{T}}\mathbf{A}^{\mathsf{T}}\mathbf{D}_{\boldsymbol{\tau}_p}\mathbf{A}\mathbb{E}(\mathbf{x}|b_x) + \mathrm{const}\\
&\overset{(a)}{=} D(b_x\|\mathrm{e}^{-f_x}) + \big(\mathbf{1}./\boldsymbol{\tau}_r\big)^{\mathsf{T}}\mathrm{var}(\mathbf{x}|b_x) - (\mathbf{s}^t)^{\mathsf{T}}\mathbf{A}\mathbb{E}(\mathbf{x}|b_x)\\
&\quad + \tfrac{1}{2}\mathbb{E}(\mathbf{x}|b_x)^{\mathsf{T}}\mathbf{D}_{\boldsymbol{\tau}_r}\mathbb{E}(\mathbf{x}|b_x) - (\mathbf{x}^t)^{\mathsf{T}}\mathbf{D}_{\boldsymbol{\tau}_r}\mathbb{E}(\mathbf{x}|b_x) + \mathrm{const}\\
&\overset{(b)}{=} D(b_x\|\mathrm{e}^{-f_x}) + \big(\mathbf{1}./\boldsymbol{\tau}_r\big)^{\mathsf{T}}\mathrm{var}(\mathbf{x}|b_x)\\
&\quad + \tfrac{1}{2}\mathbb{E}(\mathbf{x}|b_x)^{\mathsf{T}}\mathbf{D}_{\boldsymbol{\tau}_r}\mathbb{E}(\mathbf{x}|b_x) - (\mathbf{r}^t)^{\mathsf{T}}\mathbf{D}_{\boldsymbol{\tau}_r}\mathbb{E}(\mathbf{x}|b_x) + \mathrm{const}\\
&= D(b_x\|\mathrm{e}^{-f_x}) + \big(\mathbf{1}./\boldsymbol{\tau}_r\big)^{\mathsf{T}}\mathrm{var}(\mathbf{x}|b_x) + \tfrac{1}{2}\|\mathbb{E}(\mathbf{x}|b_x) - \mathbf{r}^t\|_{\boldsymbol{\tau}_r}^2\\
&\quad + \mathrm{const}\\
&= D(b_x\|\mathrm{e}^{-f_x}) + \mathbb{E}\big(\tfrac{1}{2}\|\mathbf{x} - \mathbf{r}^t\|_{\boldsymbol{\tau}_r}^2\big|b_x\big) + \mathrm{const}, \qquad (157)\\
&= \int_{\mathbb{R}^n} b_x(\mathbf{x}) \ln \frac{b_x(\mathbf{x})}{\exp(-f_x(\mathbf{x}) - \frac{1}{2}\|\mathbf{x} - \mathbf{r}^t\|_{\boldsymbol{\tau}_r}^2)}\,\mathrm{d}\mathbf{x} + \mathrm{const} \qquad (158)\\
&\overset{(c)}{=} D(b_x\|p_x) + \mathrm{const}, \qquad (159)
\end{aligned}$$

where "const" is constant with respect to b_x; line (a) used (80c); line (b) used

$$\mathbf{r}^t \triangleq \mathbf{x}^t + \mathrm{Diag}(\boldsymbol{\tau}_r)\mathbf{A}^{\mathsf{T}}\mathbf{s}^t; \qquad (160)$$

and line (c) used $p_x(\mathbf{x}) \propto \exp(-f_x(\mathbf{x}) - \frac{1}{2}\|\mathbf{x} - \mathbf{r}^t\|_{\boldsymbol{\tau}_p}^2)$. Thus, the minimizing density b_x output by (80a) is

$$b_x^{t+1}(\mathbf{x}) \propto \exp\big(-f_x(\mathbf{x}) - \tfrac{1}{2}\|\mathbf{x} - \mathbf{r}^t\|_{\boldsymbol{\tau}_r}^2\big). \qquad (161)$$

Finally, using (156) in (80c), we obtain

$$\mathbf{s}^{t+1} = (\mathbf{z}^{t+1} - \mathbf{p}^{t+1})./\boldsymbol{\tau}_p. \qquad (162)$$

Thus, we have recovered the mean updates of the original sum-product GAMP algorithm, i.e., the non-indented lines in Algorithm 4.

REFERENCES

[1] J. A. Nelder and R. W. M. Wedderburn, "Generalized linear models," *J. Roy. Stat. Soc. Ser. A*, vol. 135, pp. 370–385, May 1972.
[2] P. McCullagh and J. A. Nelder, *Generalized Linear Models*, 2nd ed. Boston, MA, USA: Chapman, 1989.
[3] A. Chambolle, R. A. De Vore, N.-Y. Lee, and B. J. Lucier, "Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 319–335, Mar. 1998.
[4] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, Nov. 2004.
[5] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2479–2493, Jul. 2009.

[6] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problem," *SIAM J. Imag. Sci.*, vol. 2, no. 1, p. 183–202, 2009.

[7] Y. E. Nesterov, "Gradient methods for minimizing composite objective function," *Center for Operations Research and Econometrics (CORE)*. Belgium, Germany: Univ. Catholique de Louvain, 2007.

[8] J. M. Bioucas-Dias and M. A. T. Figueiredo, "A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Trans. Image Process.*, vol. 16, no. 12, pp. 2992–3004, Dec. 2007.

[9] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[10] E. Esser, X. Zhang, and T. F. Chan, "A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science," *SIAM J. Imag. Sci.*, vol. 3, no. 4, pp. 1015–1046, 2010.

[11] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imag. Vis.*, vol. 40, no. 1, pp. 120–145, 2011.

[12] B. He and X. Yuan, "Convergence analysis of primal-dual algorithms for a saddle-point problem: From contraction perspective," *SIAM J. Imag. Sci.*, vol. 5, no. 1, pp. 119–149, 2012.

[13] L. A. Rademacher, "Approximating the centroid is hard," in *Proc. ACM Comput. Geometry*, 2007, pp. 302–305.

[14] N. E. Breslow and D. G. Clayton, "Approximate inference in generalized linear mixed models," *J. Amer. Statist. Assoc.*, vol. 88, no. 421, pp. 9–25, 1993.

[15] S. L. Zeger and M. R. Karim, "Generalized linear models with random effects: A Gibbs sampling approach," *J. Amer. Statist. Assoc.*, vol. 86, no. 413, pp. 79–86, 1991.

[16] D. Gamerman, "Sampling from the posterior distribution in generalized linear mixed models," *Statist. Comput.*, vol. 7, no. 1, pp. 57–68, 1997.

[17] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proc. Nat. Acad. Sci.*, vol. 106, no. 45, pp. 18914–18919, Nov. 2009.

[18] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. Motivation and construction," in *Proc. Inf. Theory Workshop*, Jan. 2010, pp. 1–5.

[19] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing II: Analysis and validation," in *Proc. Inf. Theory Workshop*, Jan. 2010, pp. 1–5.

[20] S. Rangan, "Estimation with random linear mixing, belief propagation and compressed sensing," in *Proc. Conf. Inf. Sci. Syst.*, Princeton, NJ, USA, Mar. 2010, pp. 1–6.

[21] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," in *Proc. IEEE Int. Symp. Inf. Theory*, Saint Petersburg, Russia, Jul./Aug. 2011, pp. 2174–2178.

[22] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 764–785, Feb. 2011.

[23] A. Javanmard and A. Montanari, "State evolution for general approximate message passing algorithms, with applications to spatial coupling," *Inf. Inference*, vol. 2, no. 2, pp. 115–144, 2013.

[24] S. Rangan, P. Schniter, and A. Fletcher, "On the convergence of approximate message passing with arbitrary matrices," in *Proc. IEEE ISIT*, Jul. 2014, pp. 236–240.

[25] F. Caltagirone, L. Zdeborová, and F. Krzakala, "On convergence of approximate message passing," in *Proc. IEEE ISIT*, Jul. 2014, pp. 1812–1816.

[26] A. Manoel, F. Krzakala, E. W. Tramel, and L. Zdeborová, "Swept approximate message passing for sparse estimation," in *Proc. ICML*, 2015, pp. 1123–1132.

[27] J. Vila, P. Schniter, S. Rangan, F. Krzakala, and L. Zdeborová, "Adaptive damping and mean removal for the generalized approximate message passing algorithm," in *Proc. IEEE ICASSP*, Apr. 2015, pp. 2021–2025.

[28] S. Rangan, P. Schniter, E. Riegler, A. Fletcher, and V. Cevher, "Fixed points of generalized approximate message passing with arbitrary matrices," in *Proc. ISIT*, Jul. 2013, pp. 664–668.

[29] F. Krzakala, A. Manoel, E. W. Tramel, and L. Zdeborová, "Variational free energies for compressed sensing," in *Proc. IEEE ISIT*, Jul. 2014, pp. 1499–1503.

[30] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," in *Exploring Artificial Intelligence New Millennium*. San Francisco, CA, USA: Morgan Kaufmann Publishers, 2003, pp. 239–269.

[31] A. L. Yuille and A. Rangarajan, "The concave-convex procedure (CCCP)," in *Advances in Neural Information Processing Systems*, vol. 2. Vancouver, BC, Canada: MIT Press, Dec. 2002, pp. 1033–1040.

[32] M. Ibrahimi, A. Javanmard, Y. Kanoria, and A. Montanari, "Robust max-product belief propagation," in *Proc. ASILOMAR*, 2011, pp. 43–49.

[33] T. Tanaka, "A statistical-mechanics approach to large-system analysis of CDMA multiuser detectors," *IEEE Trans. Inf. Theory*, vol. 48, no. 11, pp. 2888–2910, Nov. 2002.

[34] S. Rangan, A. K. Fletcher, and V. K. Goyal, "Asymptotic analysis of MAP estimation via the replica method and compressed sensing," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, Canada, Dec. 2009, pp. 1545–1553.

[35] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm," *IEEE Trans. Signal Process.*, vol. 45, no. 3, pp. 600–616, Mar. 1997.

[36] T. Blumensath, "Compressed sensing with nonlinear observations and related nonlinear optimization problems," *IEEE Trans. Inf. Theory*, vol. 59, no. 6, pp. 3466–3474, Jun. 2013.

[37] C. M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics). New York, NY, USA: Springer, 2006.

[38] U. S. Kamilov, V. K. Goyal, and S. Rangan, "Message-passing de-quantization with applications to compressed sensing," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6270–6281, Dec. 2012.

[39] U. S. Kamilov, S. Rangan, A. K. Fletcher, and M. Unser, "Approximate message passing with consistent parameter estimation and applications to sparse learning," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2969–2985, May 2014.

[40] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, nos. 1–2, pp. 1–305, 2008.

[41] D. Baron, S. Sarvotham, and R. G. Baraniuk, "Bayesian compressive sensing via belief propagation," *IEEE Trans. Signal Process.*, vol. 58, no. 1, pp. 269–280, Jan. 2010.

[42] M. Welling and Y. W. Teh, "Belief optimization for binary networks: A stable alternative to loopy belief propagation," in *Proc. Uncertainty Artif. Intell.*, 2001, pp. 554–561.

[43] J. Shin, "The complexity of approximating a Bethe equilibrium," *IEEE Trans. Inf. Theory*, vol. 60, no. 7, pp. 3959–3969, Jul. 2014.

[44] R. T. Rockafellar, "Monotropic programming: Descent algorithms and duality," in *Nonlinear Programming*, vol. 4. O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, Eds. New York, NY, USA: Academic, 1981, pp. 327–366.

[45] J. Nocedal and S. Wright, *Numerical Optimization*. Springer, 2006.

[46] Y. Weiss, C. Yanover, and T. Meltzer, "MAP estimation, linear programming and belief propagation with convex free energies," in *Proc. UAI*, 2007, pp. 416–425.

[47] S. Rangan, A. K. Fletcher, and V. K. Goyal, "Asymptotic analysis of MAP estimation via the replica method and applications to compressed sensing," *IEEE Trans. Inf. Theory*, vol. 58, no. 3, pp. 1902–1923, Mar. 2012.

[48] A. Dembo and O. Zeitouni, *Large Deviations Techniques and Applications*. New York, NY, USA: Springer, 1998.

[49] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 3, no. 1, pp. 123–231, 2013.

[50] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc., B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[51] P. T. Boufounos and R. G. Baraniuk, "1-bit compressive sensing," in *Proc. Conf. Inf. Sci. Sys.*, 2008, pp. 16–21.

[52] F. Krzakala and M. Mézard, F. Sausset, Y. Sun, and L. Zdeborová, "Statistical-physics-based reconstruction in compressed sensing," *Phys. Rev. X*, vol. 2, no. 2, p. 021005, 2012.

[53] J. P. Vila and P. Schniter, "Expectation-maximization Gaussian-mixture approximate message passing," *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4658–4672, Oct. 2013.

[54] J. P. Vila and P. Schniter, "An empirical-Bayes approach to recovering linearly constrained non-negative sparse signals," *IEEE Trans. Signal Process.*, vol. 62, no. 18, pp. 4689–4703, Sep. 2014.

[55] S. Som and P. Schniter, "Compressive imaging using approximate message passing and a Markov-tree prior," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3439–3448, Jul. 2012.

[56] S. Rangan, A. K. Fletcher, V. K. Goyal, and P. Schniter, "Hybrid generalized approximation message passing with applications to structured sparsity," in *Proc. IEEE Int. Symp. Inf. Theory*, Cambridge, MA, USA, Jul. 2012, pp. 1241–1245.

[57] J. T. Parker, P. Schniter, and V. Cevher, "Bilinear generalized approximate message passing—Part I: Derivation," *IEEE Trans. Inf. Theory*, vol. 62, no. 22, pp. 5839–5853, Nov. 2014.
[58] J. T. Parker, P. Schniter, and V. Cevher, "Bilinear generalized approximate message passing—Part II: Applications," *IEEE Trans. Inf. Theory*, vol. 62, no. 22, pp. 5854–5867, Nov. 2014.
[59] S. Rangan and A. K. Fletcher, "Iterative estimation of constrained rank-one matrices in noise," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2012, pp. 1246–1250.
[60] H. J. Brascamp and E. H. Lieb, "On extensions of the Brunn–Minkowski and Prékopa–Leindler theorems, including inequalities for log concave functions, and with an application to the diffusion equation," *In Inequalities*. Berlin, Germany: Springer, 2002, pp. 441–464.
[61] M. Vidyasagar, *Nonlinear Systems Analysis*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1978.
[62] R. D. Yates, "A framework for uplink power control in cellular radio systems," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 7, pp. 1341–1347, Sep. 1995.

Sundeep Rangan (M'02–SM'14–F'16) received the B.A.Sc. degree from the University ofWaterloo, Waterloo, ON, Canada, and the M.Sc. and Ph.D. degrees from the University of California, Berkeley, Berkeley, CA, USA, all in electricalengineering. He has held postdoctoral appointments with the University of Michigan, Ann Arbor, MI, USA, and Bell Labs. In 2000, he cofounded (with four others) Flarion Technologies, a spin-off of Bell Labs, that developed Flash OFDM, the first cellular OFDM data system and precursor to 4G systems including LTE and WiMAX. In 2006, Flarion was acquired by Qualcomm Technologies. He was the Director of Engineering at Qualcomm involved in OFDM infrastructure products. He joined the Department of ECE, NYU, in 2010, where he is currently an Associate Professor and the Director of NYU WIRELESS. His research interests include wireless communications, signal processing, information theory, and control theory.

Alyson K. Fletcher (S'03–M'04) received the B.S. degree in mathematics from the University of Iowa. From the University of California, Berkeley, she received the M.S. degree in electrical engineering in 2002, and the M.A. degree in mathematics and Ph.D. degree in electrical engineering, both in 2006.

Dr. Fletcher is a member of SWE, SIAM, and Sigma Xi. In 2005, she received the University of California Eugene L. Lawler Award, the Henry Luce Foundations Clare Boothe Luce Fellowship, the Soroptimist Dissertation Fellowship, and University of California Presidents Postdoctoral Fellowship. Her research interests include signal processing, information theory, machine learning, and neuroscience.

Philip Schniter (S'92–M'93–SM'05–F'14) received the B.S. and M.S. degrees in Electrical Engineering from the University of Illinois at Urbana-Champaign in 1992 and 1993, respectively, and the Ph.D. degree in Electrical Engineering from Cornell University in Ithaca, NY, in 2000.

From 1993 to 1996 he was employed by Tektronix Inc. in Beaverton, OR as a systems engineer. After receiving the Ph.D. degree, he joined the Department of Electrical and Computer Engineering at The Ohio State University, Columbus, where he is currently a Professor and a member of the Information Processing Systems (IPS) Lab. In 2008-2009 he was a Visiting Professor at Eurecom, Sophia Antipolis, France, and Supélec, Gif-sur-Yvette, France. In 2016-2017 he was a Visiting Professor at Duke University, Durham, NC. His areas of interest currently include signal processing, wireless communications, and machine learning.

Ulugbek S. Kamilov (S'11–M'15) is a Research Scientist in the Computational Sensing Team at Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA. Dr. Kamilov obtained his B.Sc. and M.Sc. in Communication Systems, and Ph.D. in Electrical Engineering from the École Polytechnique Fédérale de Lausanne (EPFL), Switzerland, in 2008, 2011, and 2015, respectively. In 2007, he was an Exchange Student at Carnegie Mellon University (CMU), Pittsburgh, PA, USA; in 2010, a Visiting Student at Massachusetts Institute of Technology (MIT), Cambridge, MA, USA; and in 2013, a Visiting Student Researcher at Stanford University, Stanford, CA, USA.