Inference With Deep Generative Priors in High Dimensions

Parthe Pandit, *Student Member, IEEE*, Mojtaba Sahraee-Ardakan, *Student Member, IEEE*, Sundeep Rangan, *Fellow, IEEE*, Philip Schniter, *Fellow, IEEE*, and Alyson K. Fletcher, *Member, IEEE*

***Abstract*—Deep generative priors offer powerful models for complex-structured data, such as images, audio, and text. Using these priors in inverse problems typically requires estimating the input and/or hidden signals in a multi-layer deep neural network from observation of its output. While these approaches have been successful in practice, rigorous performance analysis is complicated by the non-convex nature of the underlying optimization problems. This paper presents a novel algorithm, Multi-Layer Vector Approximate Message Passing (ML-VAMP), for inference in multi-layer stochastic neural networks. ML-VAMP can be configured to compute maximum a priori (MAP) or approximate minimum mean-squared error (MMSE) estimates for these networks. We show that the performance of ML-VAMP can be exactly predicted in a certain high-dimensional random limit. Furthermore, under certain conditions, ML-VAMP yields estimates that achieve the minimum (i.e., Bayes-optimal) MSE as predicted by the replica method. In this way, ML-VAMP provides a computationally efficient method for multi-layer inference with an exact performance characterization and testable conditions for optimality in the large-system limit.**

***Index Terms*—Analyzing deep neural networks, inverse problems, vector approximate message passing, stochastic neural networks, state evolution.**

Manuscript received October 16, 2019; accepted March 9, 2020. Date of publication April 8, 2020; date of current version June 8, 2020. The work of Parthe Pandit, Mojtaba Sahraee-Ardakan, and Alyson K. Fletcher was supported in part by the National Science Foundation under Grant 1254204 and Grant 1738286, and in part by the Office of Naval Research under Grant N00014-15-1-2677. The work of Sundeep Rangan was supported in part by the National Science Foundation under Grant 1116589, Grant 1302336, and Grant 1547332, and in part by the industrial affiliates of NYU WIRELESS. The work of Philip Schniter was supported in part by the National Science Foundation under Grant 1716388. Portions of this paper were presented at the IEEE International Symposium on Information Theory in 2018 [1] and 2019 [2]. *(Corresponding author: Parthe Pandit.)*

Parthe Pandit, Mojtaba Sahraee-Ardakan, and Alyson K. Fletcher are with the Department of Electrical and Computer Engineering, University of California at Los Angeles, Los Angeles, CA 90095 USA, and also with the Department of Statistics, University of California at Los Angeles, Los Angeles, CA 90095 USA (e-mail: parthepandit@ucla.edu; msahraee@ucla.edu; akfletcher@ucla.edu).

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

Philip Schniter is with the Department of Electrical and Computer Engineering, Ohio State University, Columbus, OH 43210 USA (e-mail: schniter.1@osu.edu).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/JSAIT.2020.2986321

I. Introduction

A. Inference With Deep Generative Priors

We consider inference in an L-layer stochastic neural network of the form

$$\mathbf{z}_\ell^0 = \mathbf{W}_\ell \mathbf{z}_{\ell-1}^0 + \mathbf{b}_\ell + \boldsymbol{\xi}_\ell, \quad \ell = 1, 3, \ldots, L-1, \tag{1a}$$

$$\mathbf{z}_\ell^0 = \boldsymbol{\phi}_\ell\Big(\mathbf{z}_{\ell-1}^0, \boldsymbol{\xi}_\ell\Big), \quad \ell = 2, 4, \ldots, L, \tag{1b}$$

where $\mathbf{z}_0^0$ is the network input, $\{\mathbf{z}_\ell^0\}_{\ell=1}^{L-1}$ are hidden-layer signals, and $\boldsymbol{y} := \mathbf{z}_L^0$ is the network output. The odd-indexed layers (1a) are (fully connected) affine linear layers with weights $\mathbf{W}_\ell$, biases $\mathbf{b}_\ell$, and additive noise vectors $\boldsymbol{\xi}_\ell$. The even-indexed layers (1b) involve separable and possibly nonlinear functions $\boldsymbol{\phi}_\ell$ that are randomized[1] by the noise vectors $\boldsymbol{\xi}_\ell$. By "separable," we mean that $[\boldsymbol{\phi}_\ell(\mathbf{z}, \boldsymbol{\xi})]_i = \phi_\ell(z_i, \xi_i)\ \forall i$, where ϕ_ℓ is some scalar-valued function, such as a sigmoid or ReLU, and where z_i and ξ_i represent the ith component of $\mathbf{z}$ and $\boldsymbol{\xi}$. We assume that the input $\mathbf{z}_0^0$ and noise vectors $\boldsymbol{\xi}_\ell$ are mutually independent, that each contains i.i.d. entries, and that the number of layers, L, is even. A block diagram of the network is shown in the top panel of Fig. 2. The *inference problem* is to estimate the input and hidden signals $\{\mathbf{z}_\ell\}_{\ell=0}^{L-1}$ from an observation of the network output $\boldsymbol{y}$. That is,

$$\text{Estimate } \{\mathbf{z}_\ell\}_{\ell=0}^{L-1} \text{ given } \boldsymbol{y},\ \{\mathbf{W}_{2k-1}, \mathbf{b}_{2k-1}, \boldsymbol{\phi}_{2k}\}_{k=1}^{L/2}. \tag{2}$$

For inference, we will assume that network parameters (i.e., the weights $\mathbf{W}_\ell$, biases $\mathbf{b}_\ell$, and activation functions $\boldsymbol{\phi}_\ell$) are all known, as are the distributions of the input $\mathbf{z}_0^0$ and the noise terms $\boldsymbol{\xi}_\ell$. Hence, we do *not* consider the network learning problem. The superscript "0" on $\mathbf{z}_\ell^0$ indicates that this is the "true" value of $\mathbf{z}_\ell$, to be distinguished from the estimates of $\mathbf{z}_\ell$ produced during inference denoted by $\widehat{\mathbf{z}}_\ell$.

The inference problem (2) arises in the following state-of-the-art approach to inverse problems. In general, solving an "inverse problem" means recovering some signal $\boldsymbol{x}$ from a measurement $\boldsymbol{y}$ that depends on $\boldsymbol{x}$. For example, in compressed sensing (CS) [5], the measurements are often modeled as $\boldsymbol{y} = \mathbf{A}\boldsymbol{x} + \boldsymbol{\xi}$ with known $\mathbf{A}$ and additive white Gaussian noise (AWGN) $\boldsymbol{\xi}$, and the signal is often modeled as a sparse linear combination of elements from a known dictionary, i.e.,

[1] The role of the noise $\xi_{\ell,i}$ in ϕ_ℓ is allowed to be generic (e.g., additive, multiplicative, etc.). The relationship between $z_{\ell,i}^0$ and $z_{\ell-1,i}^0$ will be modeled using the conditional density $p(z_{\ell,i}^0|z_{\ell-1,i}^0) = \int \delta(z_{\ell,i}^0 - \phi_\ell(z_{\ell-1,i}^0, \xi_{\ell,i}))p(\xi_{\ell,i})\, \mathrm{d}\xi_{\ell,i}$.

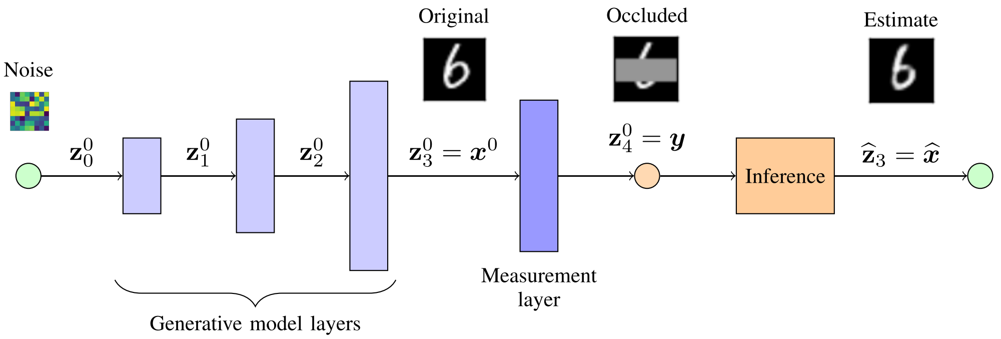

Fig. 1. Motivating example: inference for inpainting [3], [4]. An image $\boldsymbol{x}^0$ is modeled as the output of a generative model driven by white noise $\mathbf{z}_0^0$, and an occluded measurement $\boldsymbol{y}$ is generated by one additional layer. Inference is then used to recover the image $\boldsymbol{x}$ from the measurement $\boldsymbol{y}$.

$\boldsymbol{x} = \boldsymbol{\Psi}\mathbf{z}$ for some sparse coefficient vector $\mathbf{z}$. To recover $\boldsymbol{x}$, one usually computes a sparse coefficient estimate $\widehat{\mathbf{z}}$ using a LASSO-type convex optimization [6] and then uses it to form a signal estimate $\widehat{\boldsymbol{x}}$, as in

$$\widehat{\boldsymbol{x}} = \boldsymbol{\Psi}\widehat{\mathbf{z}} \quad \text{for} \quad \widehat{\mathbf{z}} = \underset{\mathbf{z}}{\operatorname{argmin}} \ \tfrac{1}{2}\|\boldsymbol{y} - \mathbf{A}\boldsymbol{\Psi}\mathbf{z}\|^2 + \lambda\|\mathbf{z}\|_1, \quad (3)$$

where $\lambda > 0$ is a tunable parameter. The CS recovery approach (3) can be interpreted as a *two-layer* version of the inference problem: the first layer implements signal generation via $\boldsymbol{x} = \boldsymbol{\Psi}\mathbf{z}$, while the second layer implements the measurement process $\boldsymbol{y} = \mathbf{A}\mathbf{z} + \boldsymbol{\xi}$. Equation (3) then performs maximum a posteriori inference (see the discussion around equation (6)) to recover estimates of $\mathbf{z}$ and $\boldsymbol{x}$.

Although CS has met with some success, it has a limited ability to exploit the complex structure of natural signals, such as images, audio, and video. This is because the model "$\boldsymbol{x} = \boldsymbol{\Psi}\mathbf{z}$ with sparse $\mathbf{z}$" is overly simplistic; it is a *one-layer* generative model. Much more sophisticated modeling is possible with multi-layer priors, as demonstrated in recent works on variational autoencoders (VAEs) [7], [8], generative adversarial networks (GANs) [9], [10], and deep image priors (DIP) [11], [12]. These models have had tremendous success in modeling richly structured data, such as images and text.

A typical application of solving an inverse problem using a deep generative model is shown in Fig. 1. This figure considers the classic problem of *inpainting* [13], for which reconstruction with DIP has been particularly successful [3], [4]. Here, a noise-like *innovation* signal $\mathbf{z}_0^0$ drives a three-layer generative network to produce an image $\boldsymbol{x}^0$. The generative network would have been trained on an ensemble of images similar to the one being estimated using, e.g., VAE or GAN techniques. The measurement process, which manifests as occlusion in the inpainting problem, is modeled using one additional layer of the network, which produces the measurement $\boldsymbol{y}$. Inference is then used to recover the image $\boldsymbol{x}^0$ (i.e., the hidden-layer signal $\mathbf{z}_3^0$) from $\boldsymbol{y}$. In addition to inpainting, this deep-reconstruction approach can be applied to other *linear* inverse problems (e.g., CS, de-blurring, and super-resolution) as well as *generalized-linear* [14] inverse problems (e.g., classification, phase retrieval, and estimation from quantized outputs). We note that the inference approach provides an alternative to designing and training a separate reconstruction network, such as in [15]–[17].

When using deterministic deep generative models, the unknown signal $\boldsymbol{x}^0$ can be modeled as $\boldsymbol{x}^0 = \mathcal{G}(\mathbf{z}_0^0)$, where $\mathcal{G}$ is a trained deep neural network and $\mathbf{z}_0^0$ is a realization of an i.i.d. random vector, typically with a Gaussian distribution. Consequently, to recover $\boldsymbol{x}^0$ from a linear-AWGN measurement of the form $\boldsymbol{y} = \mathbf{A}\boldsymbol{x}^0 + \boldsymbol{\xi}$, the compressed-sensing approach in (3) can be extended to a regularized least-squares problem [18] of the form

$$\widehat{\boldsymbol{x}} = \mathcal{G}(\widehat{\mathbf{z}}_0), \quad \widehat{\mathbf{z}}_0 := \underset{\mathbf{z}}{\operatorname{argmin}} \ \tfrac{1}{2}\|\boldsymbol{y} - \mathbf{A}\mathcal{G}(\mathbf{z})\|^2 + \lambda\|\mathbf{z}\|^2. \quad (4)$$

In practice, the optimization in (4) is solved using a gradient-based method. This approach can be straightforwardly implemented with deep-learning software packages and has been used, with excellent results, in [3], [4], [19]–[23]. The minimization (4) has also been useful in interpreting the semantic meaning of hidden signals in deep networks [24], [25]. VAEs [7], [8] and certain GANs [26] can also produce decoding networks that sample from the posterior density, and sampling methods such as Markov-chain Monte Carlo (MCMC) algorithms and Langevin diffusion [27], [28] can also be employed. We note that while the weight matrices in the motivating example in Fig. 1 are constant, during analysis we assume that they are instances of random matrices drawn from a general distribution of random matrices.

B. Analysis via Approximate Message Passing (AMP)

While reconstruction with deep generative priors has seen tremendous practical success, its performance is not fully understood. Optimization approaches such as (4) are typically non-convex and difficult to analyze. As we discuss below, most results available today only provide bounds, and these bounds are often be overly conservative (see Section I-D).

Given a network architecture and statistics on the unknown signals, fundamental information-theoretic questions include: What are the precise limits on the accuracy of estimating the hidden signals $\{\mathbf{z}_\ell^0\}_{\ell=0}^{L-1}$ from the measurements $\boldsymbol{y}$? How well do current estimation methods perform relative to these limits? Is it possible to design computationally efficient yet optimal methods?

To answer these questions, this paper considers deep inference via approximate message passing (AMP), a powerful approach for analyzing estimation problems in certain high-dimensional random settings. Since its origins in understanding linear inverse problems in compressed sensing [29], [30], AMP has been extended to an impressive range of estimation and learning tasks, including generalized linear models [31], models with parametric uncertainty [32], structured priors [33], and bilinear problems [34]. For these problems, AMP-based methods have been able to provide computationally efficient algorithms with precise high-dimensional analyses. Often, AMP approaches yield optimality guarantees in cases where all other known approaches do not. See [35] for a detailed discussion on the optimality of AMP.

C. Main Contributions

In this work, we develop a multi-layer version of AMP for inference in deep networks. The proposed approach builds on the recent vector AMP (VAMP) method of [36], which is itself closely related to expectation propagation (EP) [37], [38], expectation-consistent approximate inference (EC) [39], [40], S-AMP [41], and orthogonal AMP [42]. The proposed method is called *multi-layer VAMP*, or ML-VAMP. As will be described in detail below, ML-VAMP estimates the hidden signals in a deep network by cycling through a set of relatively simple *estimation functions* $\{\mathbf{g}_\ell^\pm\}_{\ell=0}^L$. The information flow in ML-VAMP is shown in the bottom panel of Fig. 2. The ML-VAMP method is similar to the multi-layer AMP method of [43] but can handle a more general class of matrices in the linear layers. In addition, as we will describe below, the proposed ML-VAMP algorithm can be configured for either MAP or MMSE estimation. We will call these approaches MAP-ML-VAMP and MMSE-ML-VAMP.

We establish several key results on the ML-VAMP algorithm:

- We show that, for both MAP and MMSE inference, the fixed points of the ML-VAMP algorithm correspond to stationary points of variational formulations of these estimators. This allows the interpretation of ML-VAMP as a Lagrangian algorithm with adaptive step-sizes in both cases. These findings are given in Theorems 1 and 2 and are similar to previous results for AMP [44], [45]. Section III describes these results.
- We prove that, in a certain large system limit (LSL), the behavior of ML-VAMP is exactly described by a deterministic recursion called the *state evolution* (SE). This SE analysis is a multi-layer extension of similar results [36], [46], [47] for AMP and VAMP. The SE equations enable *asymptotically exact* predictions of macroscopic behaviors of the hidden-layer estimates for *each iteration* of the ML-VAMP algorithm. This allows us to obtain error bounds even if the algorithm is run for a finite number of iterations. The SE analysis, given in Theorem 3, is the main contribution of the paper, and is discussed in Section IV.
- Since the original conference versions of this paper [1], [2], formulae for the minimum mean-squared error (MMSE) for inference in deep networks have been conjectured in [48]–[50]. As discussed in Section IV-C, these formulae are based on heuristic techniques, such as the replica method from statistical physics, and have been rigorously proven in special cases [51], [52]. Remarkably, we show that the mean-squared-error (MSE) of ML-VAMP exactly matches the predicted MMSE in certain cases.
- Using numerical simulations, we verify the predictions of the main result from Theorem 3. In particular, we show that the SE accurately predicts the MSE even for networks that are not considered large by today's standards. We also perform experiments with the MNIST handwritten digit dataset. Here we consider the inference problem using learned networks, for which the weights do not satisfy the randomness assumptions required in our analysis.

In summary, ML-VAMP provides a computationally efficient method for inference in deep networks whose performance can be exactly predicted in certain high-dimensional random settings. Moreover, in these settings, the MSE performance of ML-VAMP can match the existing predictions of the MMSE.

D. Prior Work

There has been growing interest in studying learning and inference problems in high-dimensional, random settings. One common model is the so-called *wide network*, where the dimensions of the input, hidden layers, and output are assumed to grow with a fixed linear scaling, and the weight matrices are modeled as realizations of random matrices. This viewpoint has been taken in [53]–[56], in several works that explicitly use AMP methods [43], [48], [49], [57], and in several works that use closely related random-matrix techniques [58], [59].

The existing work most closely related to ours is that by Manoel *et al.* [43], which developed a multi-layer version of the original AMP algorithm [29]. The work [43] provides a state-evolution analysis of multi-layer inference in networks with entrywise i.i.d. Gaussian weight matrices. In contrast, our results apply to the larger class of rotationally invariant matrices (see Section IV for details), which includes i.i.d. Gaussian matrices case as a special case.

Several other recent works have also attempted to characterize the performance of reconstruction using deep priors in random settings. For example, when $\mathbf{z}_0^0 \in \mathbb{R}^k$ and $\mathbf{A} \in \mathbb{R}^{m\times n}$ is a realization of an i.i.d. Gaussian matrix with $m = \Omega(kL\log n)$, Bora *et al.* [4] showed that an L-layer network $\mathcal{G}$ with ReLU activations can provide provably good reconstruction of $\boldsymbol{x}^0 \in \text{Range}(\mathcal{G})$ from measurements $\boldsymbol{y} = \mathbf{A}\boldsymbol{x}^0 + \boldsymbol{\xi}$. For the same problem, [19] and [60] show that, for $\mathbf{W}_\ell \in \mathbb{R}^{N_\ell\times N_{\ell-1}}$ generated entrywise i.i.d. Gaussian and $N_\ell = \Omega(N_{\ell-1}\log N_{\ell-1})$, one can derive bounds on reconstruction error that hold with high probability under similar conditions on m. Furthermore, they also show that the cost function of (4) has stationary points in only two disjoint regions of the $\mathbf{z}_0$ space, and both are closely related to the true solution $\mathbf{z}_0^0$. In [61], the authors use a layer-wise reconstruction scheme to prove reconstruction error bounds when $N_\ell = \Omega(N_{\ell-1})$, i.e., the network is expansive, but with a constant factor as opposed to the logarithmic factor in [60].

Our results, in comparison, provide an asymptotically exact characterization of the reconstruction error—not just bounds. Moreover, our results hold for arbitrary hidden-dimension ratios $N_\ell/N_{\ell-1}$, which can be less than, equal to, or greater than one. On the other hand, our results hold only in the large-system limit, whereas the other results above hold in the finite-dimensional regime. Nevertheless, we think that it should be possible to derive a finite-dimensional version of our analysis (in the spirit of [62]) that holds with high probability. Also, our experimental results suggest that our large-system-limit analysis is a good approximation of behavior at moderate dimensions.

Some of the material in this paper appeared in conference versions [1], [2], Theorems 1 and 3 were stated in [2], whereas Theorem 4 was stated in [1]. The current paper includes all the proofs, simulation details, and provides a unified treatment of both MAP and MMSE estimation. Additionally, Theorem 2 and its proof are new results.

II. Multi-Layer Vector Approximate Message Passing

A. Problem Formulation

We consider inference in a probabilistic setting where, in (1), $\mathbf{z}_0^0$ and $\{\boldsymbol{\xi}_\ell\}_{\ell=1}^{L}$ are modeled as random vectors with known densities. Due to the Markovian structure of $\{\mathbf{z}_\ell\}$ in (1), the posterior distribution $p(\mathbf{z}|\boldsymbol{y})$, where $\mathbf{z} := \{\mathbf{z}_0\}_{\ell=0}^{L-1}$, factorizes as

$$p(\mathbf{z}|\boldsymbol{y}) \propto p(\mathbf{z},\boldsymbol{y}) = p(\mathbf{z},\mathbf{z}_L) = p(\mathbf{z}_0)\prod_{\ell=1}^{L} p(\mathbf{z}_\ell|\mathbf{z}_{\ell-1}), \tag{5}$$

where the form of $p(\mathbf{z}_\ell|\mathbf{z}_{\ell-1})$ is determined by $\mathbf{W}_\ell$, $\mathbf{b}_\ell$, and the distribution of $\boldsymbol{\xi}_\ell$ for odd ℓ; and by $\boldsymbol{\phi}_\ell$ and the distribution of $\boldsymbol{\xi}_\ell$ for even ℓ. We will assume that $\mathbf{z}_\ell \in \mathbb{R}^{N_\ell}$, where N_ℓ can vary across the layers ℓ.

Similar to other graphical-model methods [63], we consider two forms of estimation: MAP estimation and MMSE estimation. The *maximum a priori*, or **MAP**, estimate is defined as

$$\widehat{\mathbf{z}}_{\mathsf{map}} := \underset{\mathbf{z}}{\operatorname{argmax}}\ p(\mathbf{z}|\boldsymbol{y}). \tag{6}$$

Although we will focus on MAP estimation, most of our results will apply to general M-estimators [64] of the form,

$$\widehat{\mathbf{z}}_{\mathsf{m\text{-}est}} := \underset{\mathbf{z}}{\operatorname{argmin}}\left\{\mathscr{L}_0(\mathbf{z}_0) + \sum_{\ell=1}^{L}\mathscr{L}_\ell(\mathbf{z}_\ell,\mathbf{z}_{\ell-1})\right\}$$

for loss functions $\mathscr{L}_\ell$. The MAP estimator then corresponds to loss functions $\mathscr{L}_\ell = -\ln p(\mathbf{z}_\ell|\mathbf{z}_{\ell-1})$ and $\mathscr{L}_0 = -\ln p(\mathbf{z}_0)$.

We will also consider the minimum mean-squared error, or **MMSE**, estimate, defined as

$$\widehat{\mathbf{z}}_{\mathsf{mmse}} := \mathbb{E}[\mathbf{z}|\boldsymbol{y}] = \int \mathbf{z}\,p(\mathbf{z}|\boldsymbol{y})\ \mathrm{d}\mathbf{z}. \tag{7}$$

To compute the MMSE estimate, we first compute the posterior marginals $p(\mathbf{z}_\ell|\mathbf{y})$. We will also be interested in estimating the posterior marginals $p(\mathbf{z}_\ell|\mathbf{y})$. From estimates of the posterior marginals, one can also compute other estimates, such as the mininum mean-absolute error (MMAE) estimate, i.e., the median of the posterior marginal.

Algorithm 1 Multi-Layer Vector Approximate Message Passing (ML-VAMP)

```
Require: Estimation functions $\mathbf{g}_0^+$, $\mathbf{g}_L^-$, and $\{\mathbf{g}_\ell^\pm\}_{\ell=1}^{L-1}$.
 1: Set $\boldsymbol{r}_{0\ell}^- = \mathbf{0}$ and initialize $\theta_{0\ell}^-$ for $\ell = 0, 1, \ldots, L-1$.
 2: for $k = 0, 1, \ldots, N_{\mathrm{it}} - 1$ do
 3:     // Forward Pass
 4:     $\widehat{\mathbf{z}}_{k0}^+ = \mathbf{g}_0^+(\boldsymbol{r}_{k0}^-, \theta_{k0}^+)$
 5:     $\alpha_{k0}^+ = \langle \partial\mathbf{g}_0^+(\boldsymbol{r}_{k0}^-, \theta_{k0}^+)/\partial\boldsymbol{r}_0^- \rangle$
 6:     $\boldsymbol{r}_{k0}^+ = (\widehat{\mathbf{z}}_{k0}^+ - \alpha_{k0}^+\boldsymbol{r}_{k0}^-)/(1-\alpha_{k0}^+)$
 7:     for $\ell = 1, \ldots, L-1$ do
 8:         $\widehat{\mathbf{z}}_{k\ell}^+ = \mathbf{g}_\ell^+(\boldsymbol{r}_{k\ell}^-, \boldsymbol{r}_{k,\ell-1}^+, \theta_{k\ell}^+)$
 9:         $\alpha_{k\ell}^+ = \left\langle \partial\mathbf{g}_\ell^+(\boldsymbol{r}_{k\ell}^-, \boldsymbol{r}_{k,\ell-1}^+, \theta_{k\ell}^+)/\partial\boldsymbol{r}_\ell^- \right\rangle$
10:         $\boldsymbol{r}_{k\ell}^+ = (\widehat{\mathbf{z}}_{k\ell}^+ - \alpha_{k\ell}^+\boldsymbol{r}_{k\ell}^-)/(1-\alpha_{k\ell}^+)$
11:     end for
12:
13:     // Backward Pass
14:     $\widehat{\mathbf{z}}_{k,L-1}^- = \mathbf{g}_L^-(\boldsymbol{r}_{k,L-1}^+, \theta_{kL}^-)$
15:     $\alpha_{k+1,L-1}^- = \left\langle \partial\mathbf{g}_L^-(\boldsymbol{r}_{k,L-1}^+, \theta_{kL}^-)/\partial\boldsymbol{r}_{k,L-1}^+ \right\rangle$
16:     $\boldsymbol{r}_{k+1,L-1}^- = (\widehat{\mathbf{z}}_{k,L-1}^- - \alpha_{k,L-1}^-\boldsymbol{r}_{k,L-1}^+)/(1-\alpha_{k,L-1}^-)$
17:     for $\ell = L-1, \ldots, 1$ do
18:         $\widehat{\mathbf{z}}_{k,\ell-1}^- = \mathbf{g}_\ell^-(\boldsymbol{r}_{k+1,\ell}^-, \boldsymbol{r}_{k,\ell-1}^+, \theta_{k\ell}^-)$
19:         $\alpha_{k+1,\ell-1}^- = \left\langle \partial\mathbf{g}_\ell^-(\boldsymbol{r}_{k+1,\ell}^-, \boldsymbol{r}_{k,\ell-1}^+, \theta_{k\ell}^-)/\partial\boldsymbol{r}_{\ell-1}^+ \right\rangle$
20:         $\boldsymbol{r}_{k+1,\ell-1}^- = (\widehat{\mathbf{z}}_{k,\ell-1}^- - \alpha_{k,\ell-1}^-\boldsymbol{r}_{k,\ell-1}^+)/(1-\alpha_{k,\ell-1}^-)$
21:     end for
22: end for
```

B. The ML-VAMP Algorithm

Similar to the generalized EC (GEC) [40] and generalized VAMP [65] algorithms, the ML-VAMP algorithm attempts to compute MAP or MMSE estimates using a sequence of forward-pass and backward-pass updates. The updates of the algorithm are specified in Algorithm 1. The quantities updated in the forward pass are denoted by superscript $+$, and those updated in the backward pass are denoted by superscript $-$. The notation on lines 9 and 19 means $\langle \partial\boldsymbol{f}(\boldsymbol{x}^*)/\partial\boldsymbol{x}\rangle := \frac{1}{n}\sum_{i=1}^{n}\partial f_i(x_i)/\partial x_i$ evaluated at $\boldsymbol{x} = \boldsymbol{x}^*$, where $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ acts componentwise. The update formulae can be derived similar to those for the GEC algorithm [40], using expectation-consistent approximations of the Gibbs free energy inspired by [39].

The ML-VAMP algorithm splits the estimation of $\mathbf{z} = \{\mathbf{z}_\ell\}_{\ell=0}^{L-1}$ into smaller problems that are solved by the *estimation functions* $\{\mathbf{g}_\ell^\pm\}_{\ell=1}^{L-1}$, $\mathbf{g}_0^+$ and $\mathbf{g}_L^-$. (See Figure 2, bottom panel.) As described below, the form of $\mathbf{g}_\ell^\pm$ depends on whether the goal is MAP or MMSE estimation. During the forward pass, the estimators $\mathbf{g}_\ell^+$ are invoked, whereas in the backward pass, $\mathbf{g}_\ell^-$ are invoked. Similarly, the ML-VAMP algorithm maintains two copies, $\widehat{\mathbf{z}}^+$ and $\widehat{\mathbf{z}}^-$, of the estimate of $\mathbf{z}$. For $\ell = 1, 2, \ldots, L-1$, each pair of estimators $(\mathbf{g}_\ell^+, \mathbf{g}_\ell^-)$ takes as input $\boldsymbol{r}_{\ell-1}^+$ and $\boldsymbol{r}_\ell^-$ to update the estimates $\widehat{\mathbf{z}}_\ell^+$ and $\widehat{\mathbf{z}}_{\ell-1}^-$, respectively. Similarly, $\mathbf{g}_0^+$ and $\mathbf{g}_L^-$ take inputs $\boldsymbol{r}_0^-$ and $\boldsymbol{r}_{L-1}^+$ to update

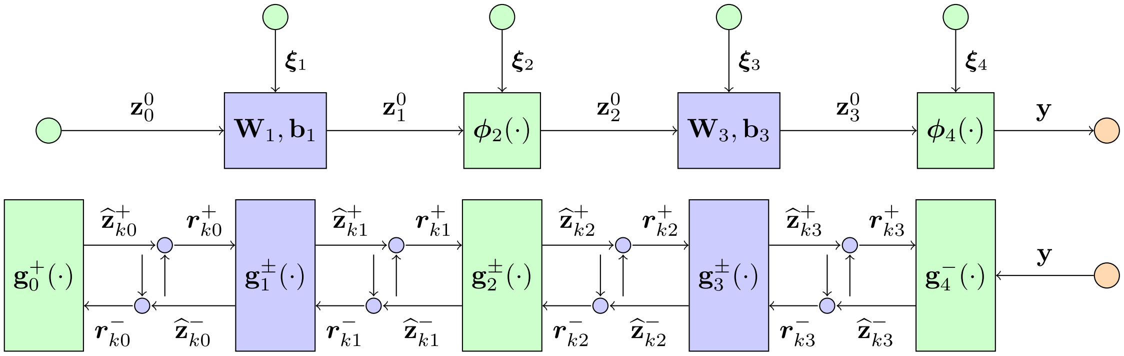
$\boldsymbol{\xi}_1$
$\boldsymbol{\xi}_2$
$\boldsymbol{\xi}_3$
$\boldsymbol{\xi}_4$
$\mathbf{z}_0^0$
$\mathbf{W}_1, \mathbf{b}_1$
$\mathbf{z}_1^0$
$\boldsymbol{\phi}_2(\cdot)$
$\mathbf{z}_2^0$
$\mathbf{W}_3, \mathbf{b}_3$
$\mathbf{z}_3^0$
$\boldsymbol{\phi}_4(\cdot)$
$\mathbf{y}$
$\mathbf{g}_0^+(\cdot)$
$\widehat{\mathbf{z}}_{k0}^+$
$\boldsymbol{r}_{k0}^+$
$\mathbf{g}_1^\pm(\cdot)$
$\widehat{\mathbf{z}}_{k1}^+$
$\boldsymbol{r}_{k1}^+$
$\mathbf{g}_2^\pm(\cdot)$
$\widehat{\mathbf{z}}_{k2}^+$
$\boldsymbol{r}_{k2}^+$
$\mathbf{g}_3^\pm(\cdot)$
$\widehat{\mathbf{z}}_{k3}^+$
$\boldsymbol{r}_{k3}^+$
$\mathbf{g}_4^-(\cdot)$
$\mathbf{y}$
$\boldsymbol{r}_{k0}^-$
$\widehat{\mathbf{z}}_{k0}^-$
$\boldsymbol{r}_{k1}^-$
$\widehat{\mathbf{z}}_{k1}^-$
$\boldsymbol{r}_{k2}^-$
$\widehat{\mathbf{z}}_{k2}^-$
$\boldsymbol{r}_{k3}^-$
$\widehat{\mathbf{z}}_{k3}^-$

Fig. 2. Top panel: Feedfoward neural network mapping an input $\mathbf{z}_0$ to output $\boldsymbol{y} = \mathbf{z}_4^0$ in the case of $L = 4$ layers. Bottom panel: ML-VAMP estimation functions $\mathbf{g}_\ell^\pm(\cdot)$ and estimation quantities $\boldsymbol{r}_{k\ell}^\pm$ and $\widehat{\mathbf{z}}_{k\ell}^\pm$ at iteration k.

$\widehat{\mathbf{z}}^0$ and $\widehat{\mathbf{z}}_{L-1}^-$, respectively. The estimation functions also take parameters $\theta_\ell^\pm$.

C. MAP and MMSE Estimation Functions: $\{\mathbf{g}_\ell^+\}$

The ML-VAMP algorithm is an iterative application of estimation functions $\mathbf{g}_\ell^\pm$ which take as input $(\mathbf{r}_\ell^-, \boldsymbol{r}_{\ell-1}^+)$ and output $(\widehat{\mathbf{z}}_\ell^+, \widehat{\mathbf{z}}_{\ell-1}^-)$. During the forward pass the output $\widehat{\mathbf{z}}_{\ell-1}^-$ is dropped whereas in the backward pass $\widehat{\mathbf{z}}_\ell^+$ is dropped. These estimation functions can take arbitrary parametric forms.

The form of the estimation functions $\{\mathbf{g}_\ell^\pm\}_{\ell=0}^{L-1}$ depends on whether the goal is to perform MAP or MMSE estimation. In either case, we restrict ourselves to the following parameterization

$$\begin{aligned} \theta_{k0}^+ &= \gamma_{k0}^-, \quad \theta_{k\ell}^+ = \left(\gamma_{k\ell}^-, \gamma_{k,\ell-1}^+\right), \\ \theta_{kL}^- &= \gamma_{k,L-1}^+ \quad \theta_{k\ell}^- = \left(\gamma_{k+1,\ell}^-, \gamma_{k,\ell-1}^+\right), \end{aligned} \tag{8}$$

where $\gamma_{k\ell}^\pm$ and $\eta_{k\ell}^\pm$ are scalars updated at iteration $k \geq 0$ and all $\ell = 0, 1, \ldots, L-1$ as follows:

$$\begin{aligned} \gamma_{k\ell}^+ &= \eta_{k\ell}^+ - \gamma_{k\ell}^-, \qquad \gamma_{k+1,\ell}^- = \eta_{k+1,\ell}^- - \gamma_{k\ell}^+, \\ \eta_{k\ell}^+ &= \gamma_{k\ell}^-/\alpha_{k\ell}^+ \qquad \eta_{k+1,\ell}^- = \gamma_{k\ell}^+/\alpha_{k+1,\ell}^-, \end{aligned} \tag{9}$$

while the updates of $\alpha_{k\ell}^\pm$ are explicitly given in lines 9 and 19 of Algorithm 1. The parameters $\gamma_{k\ell}^\pm$ and $\eta_{k\ell}^\pm$ respectively, represent estimates for precision (inverse variance) of the input $\mathbf{r}_{k\ell}^\pm$ and output $\widehat{\mathbf{z}}_{k\ell}^\pm$ to the estimation functions $\mathbf{g}_\ell^\pm$. They can also be interpreted as surrogates for curvature information (or second-order information) of the loss function. The quantities $\alpha_{k\ell}^\pm \in (0, 1)$ couple the forward and backward iterations via the so-called *Onsager correction* terms in line 10 and 20.

Given these parameters, both the MAP and MMSE estimation functions are defined from the *belief* function over $(\mathbf{z}_\ell, \mathbf{z}_{\ell-1})$:

$$\begin{aligned} b_\ell\big(\mathbf{z}_\ell, \mathbf{z}_{\ell-1} | \boldsymbol{r}_\ell^-, \boldsymbol{r}_{\ell-1}^+, \gamma_\ell^-, \gamma_{\ell-1}^+\big) &\propto p(\mathbf{z}_\ell | \mathbf{z}_{\ell-1}) \\ \times \exp\left(-\frac{\gamma_\ell^-}{2}\|\mathbf{z}_\ell - \boldsymbol{r}_\ell^-\|^2 - \frac{\gamma_{\ell-1}^+}{2}\|\mathbf{z}_{\ell-1} - \boldsymbol{r}_{\ell-1}^+\|^2\right) \end{aligned} \tag{10}$$

for $\ell = 1, 2, \ldots, L-1$. Similarly, $b_L(\mathbf{z}_L, \mathbf{z}_{L-1}) \propto p(\boldsymbol{y}|\mathbf{z}_{L-1}) \exp(-\frac{\gamma_{L-1}^+}{2}\|\mathbf{z}_{L-1} - \boldsymbol{r}_{L-1}^+\|^2)$, and $b_0(\mathbf{z}_0, \mathbf{z}_{-1}) \propto p(\mathbf{z}_0) \exp(-\frac{\gamma_0^-}{2}\|\mathbf{z}_0 - \boldsymbol{r}_0^-\|^2)$. When performing MMSE inference, we use

$$\begin{aligned} \left(\widehat{\mathbf{z}}_\ell^+, \widehat{\mathbf{z}}_{\ell-1}^-\right)_{\mathsf{mmse}} &= \mathbf{g}_{\ell,\mathsf{mmse}}^\pm\left(\boldsymbol{r}_\ell^-, \boldsymbol{r}_{\ell-1}^+; \gamma_\ell^-, \gamma_{\ell-1}^+\right) \\ &= \mathbb{E}\big[(\mathbf{z}_\ell, \mathbf{z}_{\ell-1}) | b_\ell\big], \end{aligned} \tag{11}$$

where $\mathbb{E}[\cdot | b_\ell]$ denotes expectation with respect to the (normalized) distribution b_ℓ. Similarly, for MAP inference, we use

$$\begin{aligned} (\widehat{\mathbf{z}}_\ell^+, \widehat{\mathbf{z}}_{\ell-1}^-)_{\mathsf{map}} &= \mathbf{g}_{\ell,\mathsf{map}}^\pm\left(\boldsymbol{r}_\ell^-, \boldsymbol{r}_{\ell-1}^+; \gamma_\ell^-, \gamma_{\ell-1}^+\right) \\ &= \operatorname*{arg\,max}_{\mathbf{z}_\ell, \mathbf{z}_{\ell-1}} \; b_\ell(\mathbf{z}_\ell, \mathbf{z}_{\ell-1}). \end{aligned} \tag{12}$$

Notice that (12) corresponds to the proximal operator of $-\ln p(\mathbf{z}_\ell | \mathbf{z}_{\ell-1})$. We will use "MMSE-ML-VAMP" to refer to ML-VAMP with the MMSE estimation functions (11), and "MAP-ML-VAMP" to refer to ML-VAMP with the MAP estimation functions (12).

D. Computational Complexity

A key feature of the ML-VAMP algorithm is that, for the neural network (1), the MMSE and MAP estimation functions (11) and (12) are computationally easy to compute. To see why, first recall that, for the even layers $\ell = 2, 4, \ldots L$, the map $\boldsymbol{\phi}_\ell$ in (1b) is assumed separable and the noise $\boldsymbol{\xi}_\ell$ is assumed i.i.d. As a result, $\mathbf{z}_\ell$ is conditionally independent given $\mathbf{z}_{\ell-1}$, i.e., $p(\mathbf{z}_\ell | \mathbf{z}_{\ell-1}) = \prod_i p(z_{\ell,i} | z_{\ell-1,i})$. Thus, for even ℓ, the belief function b_ℓ in (10) also factors into a product of the form $b_\ell(\mathbf{z}_\ell, \mathbf{z}_{\ell-1}) = \prod_i b_\ell(z_{\ell,i}, z_{\ell-1,i})$, implying that the MAP and MMSE versions of $\mathbf{g}_\ell^\pm$ are both coordinate-wise separable. In other words, the MAP and MMSE estimation functions can be computed using N_ℓ scalar MAP or MMSE estimators.

Next consider (1a) for $\ell = 1, 3, \ldots, L-1$, i.e., the linear layers. Assume that $\boldsymbol{\xi}_\ell \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}\nu_\ell^{-1})$ for some precision (i.e., inverse variance) $\nu_\ell > 0$. Then $p(\mathbf{z}_\ell | \mathbf{z}_{\ell-1}) \propto \frac{\nu_\ell}{2}\|\mathbf{z}_\ell - \mathbf{W}_\ell \mathbf{z}_{\ell-1} - \mathbf{b}_\ell\|^2$. In this case, the MMSE and MAP estimation functions (11) and (12) are identical, and both take the form of a standard least-squares problem. Similar to the VAMP algorithm [36], the least-squares solution—which must be recomputed at each iteration k—is can be efficiently computed using a single singular value decomposition (SVD) that is computed once, before the iterations begin. In particular, we compute the SVD

$$\mathbf{W}_\ell = \mathbf{V}_\ell \operatorname{Diag}(\boldsymbol{s}_\ell)\mathbf{V}_{\ell-1}, \tag{13}$$

where $\mathbf{V}_\ell \in \mathbb{R}^{N_\ell \times N_\ell}$ and $\mathbf{V}_{\ell-1} \in \mathbb{R}^{N_{\ell-1}\times N_{\ell-1}}$ are orthogonal and $\operatorname{Diag}(\boldsymbol{s}_\ell) \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ is a diagonal matrix that contains the singular values of $\mathbf{W}_\ell$. Let $\overline{\mathbf{b}}_\ell{:=}\mathbf{V}_\ell^\top \mathbf{b}_\ell$. Then for odd ℓ, the updates (11) and (12) both correspond to quadratic problems, which can be simplified by exploiting the rotational invariance of the ℓ_2 norm. Specifically, one can derive that

$$\begin{aligned}\widehat{\mathbf{z}}_\ell^+ &= \mathbf{g}_\ell^+(\boldsymbol{r}_\ell^-, \boldsymbol{r}_{\ell-1}^+, \gamma_\ell^-, \gamma_{\ell-1}^+)\\ &= \mathbf{V}_\ell \mathbf{G}_\ell^+\Big(\mathbf{V}_\ell^\top \boldsymbol{r}_\ell^-, \mathbf{V}_{\ell-1}\boldsymbol{r}_{\ell-1}^+, \bar{\mathbf{s}}_\ell, \overline{\mathbf{b}}_\ell, \gamma_\ell^-, \gamma_{\ell-1}^+\Big), \quad (14a)\\ \widehat{\mathbf{z}}_{\ell-1}^- &= \mathbf{g}_\ell^-(\boldsymbol{r}_\ell^-, \boldsymbol{r}_{\ell-1}^+, \gamma_\ell^-, \gamma_{\ell-1}^+)\\ &= \mathbf{V}_{\ell-1}^\mathsf{T}\mathbf{G}_\ell^-\Big(\mathbf{V}_\ell^\top \boldsymbol{r}_\ell^-, \mathbf{V}_{\ell-1}\boldsymbol{r}_{\ell-1}^+, \bar{\mathbf{s}}_\ell, \overline{\mathbf{b}}_\ell, \gamma_\ell^-, \gamma_{\ell-1}^+\Big), \quad (14b)\end{aligned}$$

where *transformed denoising functions* $\mathbf{G}_\ell^\pm(\cdot)$ are componentwise extensions of $G_\ell^\pm(\cdot)$, defined as

$$\begin{bmatrix} G_\ell^+ \\ G_\ell^- \end{bmatrix} = \begin{bmatrix} -\nu_\ell s_\ell & \gamma_\ell^- + \nu_\ell \\ \gamma_{\ell-1}^+ + \nu_\ell s_\ell^2 & -\nu_\ell s_\ell \end{bmatrix}^{-1} \begin{bmatrix} \gamma_\ell^- u_\ell + \nu_\ell \overline{b}_\ell \\ \gamma_{\ell-1}^+ u_{\ell-1} - \nu_\ell s_\ell \overline{b}_\ell \end{bmatrix} \tag{15}$$

Note that G_ℓ^+ and G_ℓ^- are functions which take inputs $(u_\ell, u_{\ell-1}, s_\ell, \overline{b}_\ell, \gamma_\ell^-, \gamma_{\ell-1}^+)$ and output the expressions on the RHS. A detailed derivation of equations (14) and (15) is given in [66, Appendix B]. Note that the argument $\bar{\mathbf{s}}_\ell$ in (14a) is N_ℓ dimensional, whereas in (14b) it is $N_{\ell-1}$ dimensional, i.e., appropriate zero-padding is applied. Keeping this subtlety in mind, we use $\bar{\mathbf{s}}_\ell$ to keep the notation simple.

From Algorithm 1, we see that each pass of the MAP-ML-VAMP or MMSE-ML-VAMP algorithm requires solving (a) scalar MAP or MMSE estimation problems for the non-linear, separable layers; and (b) least-squares problems for the linear layers. In particular, no high-dimensional integrals or high-dimensional optimizations are involved.

III. Fixed Points of ML-VAMP

Our first goal is to characterize the fixed points of Algorithm 1. To this end, let $\boldsymbol{r}_\ell^+, \boldsymbol{r}_\ell^-, \widehat{\mathbf{z}}_\ell$ with parameters $\alpha_\ell^+, \alpha_\ell^-, \gamma_\ell^+, \gamma_\ell^-, \eta_\ell$ be a fixed point of the ML-VAMP algorithm, where we have dropped the iteration subscript k. At a fixed point, we do not need to distinguish between $\widehat{\mathbf{z}}_\ell^+$ and $\widehat{\mathbf{z}}_\ell^-$, nor between η_ℓ^+ and η_ℓ^-, since the updates in (9) imply that

$$\begin{aligned}&\eta_\ell^+ = \eta_\ell^- = \gamma_\ell^+ + \gamma_\ell^- =: \eta_\ell,\\ &\alpha_\ell^+ = \frac{\gamma_\ell^-}{\eta_\ell}, \quad \alpha_\ell^- = \frac{\gamma_\ell^+}{\eta_\ell}, \quad \text{and} \quad \alpha_\ell^+ + \alpha_\ell^- = 1. \end{aligned} \tag{16}$$

Applying these relationships to lines 10 and 20 of Algorithm 1 gives

$$\widehat{\mathbf{z}}_\ell^+ = \widehat{\mathbf{z}}_\ell^- = \frac{\gamma_\ell^+ \boldsymbol{r}_\ell^+ + \gamma_\ell^- \boldsymbol{r}_\ell^-}{\gamma_\ell^+ + \gamma_\ell^-} =: \widehat{\mathbf{z}}_\ell. \tag{17}$$

A. Fixed Points of MAP-ML-VAMP and Connections to ADMM

Our first results relates the MAP-ML-VAMP updates to an ADMM-type minimization of the MAP objective (6). For this we use *variable splitting*, where we replace each variable $\mathbf{z}_\ell$ with two copies, $\mathbf{z}_\ell^+$ and $\mathbf{z}_\ell^-$. Then, we define the objective function

$$F\big(\mathbf{z}^+, \mathbf{z}^-\big) := -\ln p(\mathbf{z}_0^+) - \sum_{\ell=1}^{L-1} \ln p\big(\mathbf{z}_\ell^+ | \mathbf{z}_{\ell-1}^-\big) - \ln p\big(\mathbf{y}|\mathbf{z}_{L-1}^-\big)$$

over the variable groups $\mathbf{z}^+ := \{\mathbf{z}_\ell^+\}_{\ell=1}^{L-1}$ and $\mathbf{z}^- := \{\mathbf{z}_\ell^-\}_{\ell=1}^{L-1}$. The optimization (6) is then equivalent to

$$\min_{\mathbf{z}^+,\mathbf{z}^-} F\big(\mathbf{z}^+, \mathbf{z}^-\big) \quad \text{s.t.} \quad \mathbf{z}_\ell^+ = \mathbf{z}_\ell^-, \ \forall\, \ell = 0, \dots, L-1. \tag{18}$$

Corresponding to this constrained optimization, we define the augmented Lagrangian

$$\begin{aligned}\mathcal{L}\big(\mathbf{z}^+, \mathbf{z}^-, \mathbf{s}\big) &= F\big(\mathbf{z}^+, \mathbf{z}^-\big)\\ &\quad + \sum_{\ell=0}^{L-1} \eta_\ell \mathbf{s}_\ell^\mathsf{T}\big(\mathbf{z}_\ell^+ - \mathbf{z}_\ell^-\big) + \frac{\eta_\ell}{2}\big\|\mathbf{z}_\ell^+ - \mathbf{z}_\ell^-\big\|^2, \end{aligned} \tag{19}$$

where $\mathbf{s} := \{\mathbf{s}_\ell\}$ is a set of dual parameters, $\gamma_\ell^\pm > 0$ are weights, and $\eta_\ell = \gamma_\ell^+ + \gamma_\ell^-$. Now, for $\ell = 1, \dots, L-2$, define

$$\begin{aligned}\mathcal{L}_\ell\big(\mathbf{z}_{\ell-1}^-, \mathbf{z}_\ell^+; \mathbf{z}_{\ell-1}^+, \mathbf{z}_\ell^-, \mathbf{s}_{\ell-1}, \mathbf{s}_\ell\big) &:= -\ln p(\mathbf{z}_\ell^+|\mathbf{z}_{\ell-1}^-) + \eta_\ell \mathbf{s}_\ell^\mathsf{T}\mathbf{z}_\ell^+\\ &\quad - \eta_{\ell-1}\mathbf{s}_{\ell-1}^\mathsf{T}\mathbf{z}_{\ell-1}^- + \tfrac{\gamma_{\ell-1}^+}{2}\big\|\mathbf{z}_{\ell-1}^- - \mathbf{z}_{\ell-1}^+\big\|^2 + \tfrac{\gamma_\ell^-}{2}\big\|\mathbf{z}_\ell^+ - \mathbf{z}_\ell^-\big\|^2,\end{aligned}$$

which represents the terms in the Lagrangian $\mathcal{L}(\cdot)$ in (19) that contain $\mathbf{z}_{\ell-1}^-$ and $\mathbf{z}_\ell^+$. Similarly, define $\mathcal{L}_0(\cdot)$ and $\mathcal{L}_{L-1}(\cdot)$ using $p(\mathbf{z}_0^+)$ and $p(\mathbf{y}|\mathbf{z}_{L-1}^+)$, respectively. One can then verify that

$$\mathcal{L}\big(\mathbf{z}^+, \mathbf{z}^-, \mathbf{s}\big) = \sum_{\ell=0}^{L-1} \mathcal{L}_\ell\big(\mathbf{z}_{\ell-1}^-, \mathbf{z}_\ell^+; \mathbf{z}_{\ell-1}^+, \mathbf{z}_\ell^-, \mathbf{s}_{\ell-1}, \mathbf{s}_\ell\big).$$

Theorem 1 (MAP-ML-VAMP): Consider the iterates of Algorithm 1 with MAP estimation functions (12) for fixed $\gamma_\ell^\pm > 0$. Suppose lines 9 and 19 are replaced with fixed values $\alpha_{k\ell}^\pm = \alpha_\ell^\pm \in (0, 1)$ from (16). Let $\mathbf{s}_{k\ell}^- := \alpha_{k\ell}^+(\widehat{\mathbf{z}}_{k-1,\ell}^- - \mathbf{r}_{k\ell}^-)$ and $\mathbf{s}_{k\ell}^+ := \alpha_{k\ell}^-(\mathbf{r}_{k\ell}^+ - \widehat{\mathbf{z}}_{k\ell}^+)$. Then, for $\ell = 0, \dots, L-1$, the forward pass iterations satisfy

$$\_\_, \widehat{\mathbf{z}}_{k\ell}^+ = \operatorname*{arg\,min}_{(\mathbf{z}_{\ell-1}^-, \mathbf{z}_\ell^+)} \mathcal{L}_\ell\Big(\mathbf{z}_{\ell-1}^-, \mathbf{z}_\ell^+; \widehat{\mathbf{z}}_{k,\ell-1}^+, \widehat{\mathbf{z}}_{k-1,\ell}^-, \mathbf{s}_{k,\ell-1}^+, \mathbf{s}_{k\ell}^-\Big) \tag{20a}$$

$$\mathbf{s}_{k\ell}^+ = \mathbf{s}_{k\ell}^- + \alpha_\ell^+\Big(\widehat{\mathbf{z}}_{k\ell}^+ - \widehat{\mathbf{z}}_{k-1,\ell}^-\Big), \tag{20b}$$

whereas the backward pass iterations satisfy

$$\widehat{\mathbf{z}}_{k,\ell-1}^-, \_\_ = \operatorname*{arg\,min}_{(\mathbf{z}_{\ell-1}^-, \mathbf{z}_\ell^+)} \mathcal{L}_\ell\Big(\mathbf{z}_{\ell-1}^-, \mathbf{z}_\ell^+; \widehat{\mathbf{z}}_{k,\ell-1}^+, \widehat{\mathbf{z}}_{k\ell}^-, \mathbf{s}_{k,\ell-1}^+, \mathbf{s}_{k+1,\ell}^-\Big) \tag{21a}$$

$$\mathbf{s}_{k+1,\ell-1}^- = \mathbf{s}_{k,\ell-1}^+ + \alpha_{\ell-1}^-\Big(\widehat{\mathbf{z}}_{k,\ell-1}^+ - \widehat{\mathbf{z}}_{k,\ell-1}^-\Big). \tag{21b}$$

Further, any fixed point of Algorithm 1 corresponds to a critical point of the Lagrangian (19).

Proof: See Appendix C. ■

Theorem 1 shows that the fixed-$\{\alpha_\ell^\pm\}$ version of ML-VAMP is an ADMM-type algorithm for solving the optimization problem (18). In the case that $\alpha_\ell^+ = \alpha_\ell^-$, this algorithm is known as the Peaceman-Rachford Splitting variant of ADMM and its convergence has been studied extensively; see [67, eq. (3)] and [68], and the references therein. Different from ADMM, the full ML-VAMP algorithm adaptively updates $\{\alpha_{k\ell}^\pm\}$ in a way that exploits the local curvature of the objective in (12). Note that, in (20a) and (21a), the "_" notation means that we compute the joint minimizers over $(\mathbf{z}_{\ell-1}^+, \mathbf{z}_\ell^+)$, but only use one of them at a time for the update step.

B. Fixed Points of MMSE-ML-VAMP and Connections to Free-Energy Minimization

Recall that $\mathbf{z} := \{\mathbf{z}_\ell\}_{\ell=0}^{L-1}$ and let $\mathcal{B}$ denote the set of density functions $b(\mathbf{z})$ factorizable as $f_0(\mathbf{z}_0) f_L(\mathbf{z}_{L-1}) \prod_{\ell=1}^{L-1} f_\ell(\mathbf{z}_\ell, \mathbf{z}_{\ell-1})$. Notice that the true posterior $p(\mathbf{z}|\mathbf{y})$ from (5) belongs to this set. Essentially, this $\mathcal{B}$ captures the chain structure of the factor graph visible in the top panel of Fig. 2. For chain-structured (and, more generally, tree-structured) graphs, one can express any $b \in \mathcal{B}$ as [69] (see also [70, Sec. III-C] for a succinct description)

$$b(\mathbf{z}) = \frac{\prod_{\ell=1}^{L-1} f_\ell(\mathbf{z}_\ell, \mathbf{z}_{\ell-1})}{\prod_{\ell=1}^{L-2} q_\ell(\mathbf{z}_\ell)}, \tag{22}$$

where $\{f_\ell(\mathbf{z}_\ell, \mathbf{z}_{\ell-1})\}$ and $\{q_\ell(\mathbf{z}_\ell)\}$ are marginal density functions of $b(\mathbf{z})$. As marginal densities, they must satisfy the consistent-marginal equations

$$\begin{aligned} b(\mathbf{z}_\ell) &= \int f_\ell(\mathbf{z}_\ell, \mathbf{z}_{\ell-1})\, \mathrm{d}\mathbf{z}_{\ell-1} \\ &= q_\ell(\mathbf{z}_\ell) \\ &= \int f_{\ell+1}(\mathbf{z}_{\ell+1}, \mathbf{z}_\ell)\, \mathrm{d}\mathbf{z}_{\ell+1}, \quad \forall\, \ell = 1 \ldots L-1. \end{aligned} \tag{23}$$

Because $p(\mathbf{z}|\mathbf{y}) \in \mathcal{B}$, we can express it using variational optimization as

$$p(\mathbf{z}|\mathbf{y}) = \arg\min_{b \in \mathcal{B}} D_{\mathsf{KL}}(b(\mathbf{z}) \| p(\mathbf{z}|\mathbf{y})), \tag{24}$$

where $D_{\mathsf{KL}}(b(\mathbf{z}) \| p(\mathbf{z}|\mathbf{y})) := \int b(\mathbf{z}) \ln \frac{b(\mathbf{z})}{p(\mathbf{z}|\mathbf{y})}\, \mathrm{d}\mathbf{z}$ is the KL divergence. Plugging $b(\mathbf{z})$ from (22) into (24), we obtain

$$\begin{aligned} p(\mathbf{z}|\mathbf{y}) = \arg\min_{b \in \mathcal{B}} \Bigg\{ &\sum_{\ell=1}^{L} D_{\mathsf{KL}}(f_\ell(\mathbf{z}_\ell, \mathbf{z}_{\ell-1}) \| p(\mathbf{z}_\ell | \mathbf{z}_{\ell-1})) \\ &+ \sum_{\ell=0}^{L-1} h(q_\ell(\mathbf{z}_\ell)) \Bigg\} \quad \text{subject to (23),} \end{aligned} \tag{25}$$

where $h(q_\ell(\mathbf{z}_\ell)) := -\int q_\ell(\mathbf{z}_\ell) \ln q_\ell(\mathbf{z}_\ell)\, \mathrm{d}\mathbf{z}_\ell$ is the differential entropy of q_ℓ. The cost function in (25) is often called the Bethe free energy [69]. In summary, because $\mathcal{B}$ is tree-structured, Bethe-free-energy minimization yields the exact posterior distribution [69].

The constrained minimization (25) is computationally intractable, because both the optimization variables $\{f_\ell, q_\ell\}$ and the pointwise linear constraints (23) are infinite dimensional. Rather than solving for the exact posterior, we might instead settle for an approximation obtained by relaxing the marginal constraints (23) to the following moment-matching conditions, for all $\ell = 0, 1, \ldots L-1$:

$$\begin{aligned} &\mathbb{E}\big[\mathbf{z}_\ell | q_\ell\big] = \mathbb{E}\big[\mathbf{z}_\ell | f_\ell\big], \mathbb{E}[\|\mathbf{z}_\ell\|^2 | q_\ell] = \mathbb{E}\Big[\|\mathbf{z}_\ell\|^2 | f_\ell\Big], \\ &\mathbb{E}\big[\mathbf{z}_\ell | q_\ell\big] = \mathbb{E}\big[\mathbf{z}_\ell | f_{\ell+1}\big], \mathbb{E}[\|\mathbf{z}_\ell\|^2 | q_\ell] = \mathbb{E}\Big[\|\mathbf{z}_\ell\|^2 | f_{\ell+1}\Big]. \end{aligned} \tag{26}$$

This approach is known as expectation-consistent (EC) approximate inference [39]. Because the constraints on f_ℓ and q_ℓ in (26) are finite dimensional, standard Lagrangian-dual methods can be used to compute the optimal solution. Thus, the EC relaxation of the Bethe free energy minimization problem (25), i.e.,

$$\begin{aligned} \min_{f_\ell} \max_{q_\ell} \Bigg\{ &\sum_{\ell=1}^{L-1} D_{\mathsf{KL}}(f_\ell(\mathbf{z}_\ell, \mathbf{z}_{\ell-1}) \| p(\mathbf{z}_\ell | \mathbf{z}_{\ell-1})) \\ &+ \sum_{\ell=0}^{L-1} h(q_\ell(\mathbf{z}_\ell)) \Bigg\} \quad \text{subject to (26),} \end{aligned} \tag{27}$$

yields a tractable approximation to $p(\mathbf{z}|\mathbf{y})$.

We now establish an equivalence between the fixed points of the MMSE-ML-VAMP algorithm and the first-order stationary points of (27). The statement of the theorem uses the belief functions b_ℓ defined in (10).

Theorem 2 (MMSE-ML-VAMP): Consider a fixed point $(\{\boldsymbol{r}_\ell^\pm\}, \{\widehat{\mathbf{z}}_\ell\}, \{\gamma_\ell^\pm\})$ of Algorithm 1 with MMSE estimation functions (11). Then $\{\gamma_\ell^+ \boldsymbol{r}_\ell^+, \frac{\gamma_\ell^+}{2}, \gamma_\ell^- \boldsymbol{r}_\ell^-, \frac{\gamma_\ell^-}{2}\}$, are Lagrange multipliers for (26) such that KKT conditions are satisfied for the problem (27) at primal solutions $\{f_\ell^*, q_\ell^*\}$. Furthermore, the marginal densities take the form $f_\ell^*(\cdot) \propto b_\ell(\cdot | \boldsymbol{r}_\ell^-, \boldsymbol{r}_{\ell-1}^+, \gamma_\ell^-, \gamma_\ell^-, \gamma_{\ell-1}^+)$ and $q_\ell^* = \mathcal{N}(\widehat{\mathbf{z}}_\ell, \boldsymbol{I}/\eta_\ell)$, with $\widehat{\mathbf{z}}_\ell$ and η_ℓ given in (16)-(17).

Proof: See Appendix C. ■

The above result shows that MMSE-ML-VAMP is essentially an algorithm to iteratively solve for the parameters $(\{\boldsymbol{r}_\ell^\pm\}, \{\widehat{\mathbf{z}}_\ell\}, \{\gamma_\ell^\pm\})$ that characterize the EC fixed points. Importantly, $q_\ell^*(\mathbf{z}_\ell)$ and $f^*(\mathbf{z}_\ell, \mathbf{z}_{\ell-1})$ serve as an approximate marginal posteriors for $\mathbf{z}_\ell$ and $(\mathbf{z}_\ell, \mathbf{z}_{\ell-1})$. This enables us to not only compute the MMSE estimate (i.e., posterior mean), but also other estimates like the MMAE estimate (i.e., the posterior median), or quantiles of the marginal posteriors. Remarkably, in certain cases, these approximate marginal-posterior statistics become exact. This is one of the main contributions of the next section.

IV. Analysis in the Large-System Limit

A. LSL Model

In the previous section, we established that, for any set of deterministic matrices $\{\mathbf{W}_\ell\}$, MAP-ML-VAMP solves the MAP problem and MMSE-ML-VAMP solves the EC variational inference problem as the iterations $k \to \infty$. In this section, we extend the analysis of [36], [46] to the rigorously study the behavior of ML-VAMP at any iteration k for classes of random matrices $\{\mathbf{W}_\ell\}$ in a certain large-system limit (LSL). The model is described in the following set of assumptions.

System model: We consider a sequence of systems indexed by N. For each N, let $\mathbf{z}_\ell = \mathbf{z}_\ell^0(N) \in \mathbb{R}^{N_\ell(N)}$ be "true" vectors generated by neural network (1) for layers $\ell = 0, \ldots, L$, such that layer widths satisfy $\lim_{N\to\infty} N_\ell(N)/N = \beta_\ell \in (0,\infty)$. Also, let the weight matrices $\mathbf{W}_\ell$ in (1a) each have an SVD given by (13), where $\{\mathbf{V}_\ell\}$ are drawn uniformly from the set of orthogonal matrices in $\mathbb{R}^{N_\ell\times N_\ell}$ and independent across ℓ. The distribution on the singular values $\mathbf{s}_\ell$ will be described below.

Similar to the VAMP analysis [36], the assumption here is that weight matrices $\mathbf{W}_\ell$ are rotationally invariant, meaning that $\mathbf{V}\mathbf{W}_\ell$ and $\mathbf{W}_\ell\mathbf{V}$ are distributed identically to $\mathbf{W}_\ell$. Gaussian i.i.d. $\mathbf{W}_\ell$ as considered in the original ML-AMP work of [43] satisfy this rotationally invariant assumption, but the rotationally invariant model is more general. In particular, as described in [36], the model can have arbitrary coniditoning which is known to be a major failure mechanism of AMP methods.

ML-VAMP algorithm: We assume that we generate estimates $\widehat{\mathbf{z}}_{k\ell}^{\pm}$ from the ML-VAMP algorithm, Algorithm 1. Our analysis will apply to general estimation functions, $\mathbf{g}_\ell(\cdot)$, not necessarily the MAP or MMSE estimators. However, we require two technical conditions: for the non-linear estimators, $\mathbf{g}_\ell^{\pm}$ for $\ell = 2,4,\ldots L-2$, and $\mathbf{g}_0^+$, $\mathbf{g}_L^-$ act componentwise. Further, these estimators and their derivatives $\frac{\partial \mathbf{g}_\ell^+}{\partial z_\ell^-}, \frac{\partial \mathbf{g}_\ell^-}{\partial z_{\ell-1}^+}, \frac{\partial \mathbf{g}_0^+}{\partial z_0^-}, \frac{\partial \mathbf{g}_L^-}{\partial z_{L-1}^+}$ are uniformly Lipschitz continuous. The technical definition of uniformly Lipschitz continuous is given in Appendix A. For the linear layers, $\ell = 1,3,\ldots L-1$, we assume we apply estimators $\mathbf{g}_\ell^{\pm}$ of the form (14) where $\mathbf{G}_\ell^{\pm}$ act componentwise. Further, $\mathbf{G}_\ell^{\pm}$ along with its derivatives are uniformly Lipschitz continuous. We also assume that the activation functions $\boldsymbol{\phi}_\ell$ in equation (1b) are componentwise separable and Lipschitz continuous. To simplify the analysis, we will also assume the estimation function parameters $\theta_{k\ell}^{\pm}$ converge to fixed limits,

$$\lim_{N\to\infty} \theta_{k\ell}^{\pm}(N) = \overline{\theta}_{k\ell}^{\pm}, \tag{28}$$

for values $\overline{\theta}_{k\ell}^{\pm}$. Importantly, in this assumption, we assume that the limiting parameter values $\overline{\theta}_{k\ell}^{\pm}$ are fixed and not data dependent. However, data dependent parameters can also be modeled [36].

Distribution of the components: We follow the framework of Bayati-Montanari and describe the statistics on the unknown quantities via their empirical convergence – see Appendix A. For $\ell = 1,3,\ldots L-1$, define $\overline{\mathbf{b}}_\ell := \mathbf{V}_\ell^{\mathsf{T}}\mathbf{b}_\ell$ and $\overline{\boldsymbol{\xi}}_\ell := \mathbf{V}_\ell^{\mathsf{T}}\boldsymbol{\xi}_\ell$. We assume that the sequence of true vectors $\mathbf{z}_0^0$, singular values $\boldsymbol{s}_\ell$, bias vectors $\overline{\mathbf{b}}_\ell$, and noise realizations $\overline{\boldsymbol{\xi}}_\ell$ empirically converge as

$$\left\{z_{0,n}^0\right\} \overset{PL(2)}{=} Z_0^0, \quad \left\{\xi_{\ell,n}\right\} \overset{PL(2)}{=} \Xi_\ell, \qquad \forall\, \ell \text{ even}, \tag{29a}$$

$$\left\{(s_{\ell,n}, \overline{b}_{\ell,n}, \overline{\xi}_{\ell,n})\right\} \overset{PL(2)}{=} (S_\ell, \overline{B}_\ell, \overline{\Xi}_\ell), \qquad \forall\, \ell \text{ odd}, \tag{29b}$$

to random variables $Z_0^0, \Xi_\ell, S_\ell, \overline{B}_\ell, \overline{\Xi}_\ell$. We will also assume that the singular values are bounded, i.e., $s_{\ell,n} < S_{\ell,\max}\ \forall n$. Also, the initial vectors $\boldsymbol{r}_{0\ell}^-$ converge as,

$$\begin{aligned} \left\{\left[\boldsymbol{r}_{0\ell}^- - \mathbf{z}_\ell^0\right]_n\right\} &\overset{PL(2)}{=} Q_{0\ell}^-, \quad \ell = 0,2,\ldots,L, \\ \left\{\left[\mathbf{V}_\ell^\top\left(\boldsymbol{r}_{0\ell}^- - \mathbf{z}_\ell^0\right)\right]_n\right\} &\overset{PL(2)}{=} Q_{0\ell}^-, \quad \ell = 1,3,\ldots,L-1, \end{aligned} \tag{30}$$

where $(Q_{0\ell}^-, Q_{1\ell}^-, \ldots Q_{L-1,\ell}^-)$ is jointly Gaussian independent of Z_0^0, $\{\Xi_\ell\}$, $\{S_\ell, \overline{B}_\ell, \overline{\Xi}_\ell\}$.

State evolution: Under the above assumptions, our main result is to show that the asymptotic distribution of the quantities from ML-VAMP algorithm converge to certain distributions. The distributions are described by a set of deterministic parameters $\{\mathbf{K}_{k\ell}^+, \tau_{k\ell}^-, \overline{\alpha}_{k\ell}^{\pm}, \overline{\gamma}_{k\ell}^{\pm}, \overline{\eta}_{k\ell}^{\pm}\}$. The evolve according to a scalar recursion called the state evolution (SE), given in Algorithm 2 in Appendix B. We assume $\overline{\alpha}_{k\ell}^{\pm} \in (0,1)$ for all iterations k and $\ell = 0,1,\ldots L-1$.

B. SE Analysis in the LSL

Under these assumptions, we can now state our main result. Let $\mathbb{S}^d$ denote the space of symmetric positive definite matrices in $\mathbb{R}^{d\times d}$. The deterministic quantities $\{\mathbf{K}_{k\ell}^+, \tau_{k\ell+1}^-, \overline{\alpha}_{k\ell}^{\pm}, \overline{\gamma}_{k\ell}^{\pm}, \overline{\eta}_{k\ell}^{\pm}\}_{\ell=0}^{L-1}$ referenced in the theorem below are defined in an iteration called the State Evolution given in Algorithm 2 (see Appendix B of Supplementary materials).

Theorem 3: Consider the system under the above assumptions. There exist deterministic parameters $\{\mathbf{K}_{k\ell}^+,\ \tau_{k\ell+1}^-,\ \overline{\alpha}_{k\ell}^{\pm},\ \overline{\gamma}_{k\ell}^{\pm},\ \overline{\eta}_{k\ell}^{\pm}\}_{\ell=0}^{L-1}$ with $\mathbf{K}_{k\ell} \in \mathbb{S}^2$, $\tau_{k\ell}^- > 0$, $\overline{\gamma}_{k\ell}^{\pm} > 0$, $\overline{\eta}_{k\ell}^{\pm} > 0$, $\overline{\alpha}_{k\ell} \in (0,1)$ such that the following convergence holds. For any componentwise pseudo-Lipschitz function $\boldsymbol{\psi}$ of order 2, iteration index k, and layer index $\ell = 2,4,\ldots L-2$,

$$\begin{aligned} &\lim_{N\to\infty}\left\langle \boldsymbol{\psi}\left(\mathbf{z}_{\ell-1}^0, \widehat{\mathbf{z}}_{k,\ell-1}^-, \widehat{\mathbf{z}}_{k\ell}^+\right)\right\rangle \overset{a.s.}{=} \\ &= \mathbb{E}\Big[\psi\Big(\mathsf{A}, g_\ell^-\Big(\mathsf{C} + \phi_\ell(\mathsf{A}, \Xi_\ell), \mathsf{B} + \mathsf{A}, \overline{\gamma}_{k\ell}^-, \overline{\gamma}_{k,\ell-1}^+\Big), \\ &\qquad g_\ell^+(\mathsf{C} + \phi_\ell(\mathsf{A}, \Xi_\ell), \mathsf{B} + \mathsf{A}, \overline{\gamma}_{k\ell}^-, \overline{\gamma}_{k,\ell-1}^+)\Big)\Big], \end{aligned} \tag{31}$$

$$\lim_{N\to\infty}\left\langle \boldsymbol{\psi}\left(\mathbf{z}_0^0, \widehat{\mathbf{z}}_{k0}^+\right)\right\rangle \overset{a.s.}{=} \mathbb{E}\Big[\psi\Big(Z_0^0, g_0^+\Big(\mathsf{F} + Z_0^0, \overline{\gamma}_0^-\Big)\Big)\Big], \tag{32}$$

$$\lim_{N\to\infty}\left\langle \boldsymbol{\psi}\left(\mathbf{z}_{L-1}^0, \widehat{\mathbf{z}}_{k,L-1}^-\right)\right\rangle \overset{a.s.}{=} \mathbb{E}\psi\big(\mathsf{D}, g_L^-(\mathsf{E} + \mathsf{D}, \overline{\gamma}_{L-1}^+)\big), \tag{33}$$

where $(\mathsf{A},\mathsf{B}) \sim \mathcal{N}(0, \mathbf{K}_{k,\ell-1}^+)$ and $\mathsf{C} \sim \mathcal{N}(0, \tau_{k\ell}^-)$ are mutually independent and independent of Ξ_ℓ; $(\mathsf{D},\mathsf{E}) \sim \mathcal{N}(0, \mathbf{K}_{k,L-1}^+)$ is independent of Ξ_L and $\mathsf{F} \sim \mathcal{N}(0, \tau_{k0}^-)$ is independent of Z_0^0.

Similarly for any layer index $\ell = 1,3,\ldots,L-1$, we have

$$\begin{aligned} &\lim_{N\to\infty}\left\langle \boldsymbol{\psi}\left(\mathbf{V}_{\ell-1}\mathbf{z}_{\ell-1}^0, \mathbf{V}_{\ell-1}\widehat{\mathbf{z}}_{k,\ell-1}^-, \mathbf{V}_\ell^\top\widehat{\mathbf{z}}_{k\ell}^+\right)\right\rangle \overset{a.s.}{=} \\ &= \mathbb{E}\Big[\psi\Big(\mathsf{A}, G_\ell^-\Big(\mathsf{C} + \mathsf{D}, \mathsf{B} + \mathsf{A}, S_\ell, \overline{B}_\ell, \overline{\gamma}_{k\ell}^-, \overline{\gamma}_{k,\ell-1}^+\Big), \\ &\qquad G_\ell^+\Big(\mathsf{C} + \mathsf{D}, \mathsf{B} + \mathsf{A}, S_\ell, \overline{B}_\ell, \overline{\gamma}_{k\ell}^-, \overline{\gamma}_{k,\ell-1}^+\Big)\Big)\Big], \end{aligned} \tag{34}$$

where $(\mathsf{A},\mathsf{B}) \sim \mathcal{N}(0, \mathbf{K}_{k,\ell-1}^+)$ and $\mathsf{C} \sim \mathcal{N}(0, \tau_{k\ell}^-)$ are mutually independent and independent of $(S_\ell, \overline{B}_\ell, \overline{\Xi}_\ell)$, and $\mathsf{D} = S_\ell\mathsf{A} + \overline{B}_\ell + \overline{\Xi}_\ell$.

Furthermore, if $\overline{\gamma}_{k\ell}^{\pm}, \overline{\eta}_{k\ell}^{\pm}$, are defined analogous to (9) using $\overline{\alpha}_{k\ell}^{\pm}$, then for all ℓ,

$$\lim_{N\to\infty}\left(\alpha_{k,\ell}^{\pm}, \gamma_{k,\ell}^{\pm}, \eta_{k,\ell}^{\pm}\right) \overset{a.s.}{=} \left(\overline{\alpha}_{k,\ell}^{\pm}, \overline{\gamma}_{k,\ell}^{\pm}, \overline{\eta}_{k,\ell}^{\pm}\right). \tag{35}$$

Proof: See Appendix D. ■

The key value of Theorem 3 is that we can *exactly characterize* the asymptotic joint distribution of the true vectors $\mathbf{z}^0_\ell$ and the ML-VAMP estimates $\widehat{\mathbf{z}}^{\pm}_{k\ell}$. The asymptotic joint distribution, can be used to compute various key quantities. For example, suppose we wish to compute the mean squared error (MSE). Let $\psi(z^0, \widehat{\mathbf{z}}) = (z^0 - \widehat{\mathbf{z}})^2$, whereby $\langle \boldsymbol{\psi}(\mathbf{z}^0_\ell, \widehat{\mathbf{z}}^-_\ell) \rangle = \frac{1}{N} \|\mathbf{z}^0_\ell - \widehat{\mathbf{z}}^-_\ell\|^2$. Observe that ψ is a pseudo-Lipschitz function of order 2, whereby we can apply Theorem 3. Using (31), we get the asymptotic MSE on the k^{th}-iteration estimates for $\ell = 2, 4, \dots L-2$:

$$\lim_{N_{\ell-1}\to\infty} \tfrac{1}{N_{\ell-1}} \left\| \widehat{\mathbf{z}}^-_{k,\ell-1} - \mathbf{z}^0_{\ell-1} \right\|^2 \overset{a.s.}{=}$$

$$= \mathbb{E}\left[\left(g^-_\ell \left(\mathsf{C} + \phi_\ell(\mathsf{A}, \Xi_\ell), \mathsf{B} + \mathsf{A}, \overline{\gamma}^-_{k\ell}, \overline{\gamma}^+_{k,\ell-1} \right) - \mathsf{A} \right)^2 \right],$$

$$\lim_{N_\ell\to\infty} \tfrac{1}{N_\ell} \left\| \widehat{\mathbf{z}}^+_{k\ell} - \mathbf{z}^0_\ell \right\|^2 \overset{a.s.}{=}$$

$$= \quad \mathbb{E}\Big[\Big(g^+_\ell \left(\mathsf{C} + \phi_\ell(\mathsf{A}, \Xi_\ell), \mathsf{B} + \mathsf{A}, \overline{\gamma}^-_{k\ell}, \overline{\gamma}^+_{k,\ell-1} \right)$$

$$- \phi_\ell(\mathsf{A}, \Xi_\ell))^2 \Big],$$

where we used the fact that ϕ_ℓ is pseudo-Lipschitz of order 2, and $\mathbf{z}^0_\ell = \phi_\ell(\mathbf{z}^0_{\ell-1}, \boldsymbol{\xi}_\ell)$ from (1b). Similarly, using (34), we get the kth-iteration MSE for $\ell = 1, 3, \dots L-1$:

$$\lim_{N_{\ell-1}\to\infty} \tfrac{1}{N_{\ell-1}} \left\| \widehat{\mathbf{z}}^-_{k,\ell-1} - \mathbf{z}^0_{\ell-1} \right\|^2 = \tfrac{1}{N_{\ell-1}} \left\| \mathbf{V}_{\ell-1} \left(\widehat{\mathbf{z}}^-_{k,\ell-1} - \mathbf{z}^0_{\ell-1} \right) \right\|^2$$

$$\overset{a.s.}{=} \mathbb{E}\left[\left(G^-_\ell \left(\mathsf{C} + \mathsf{D}, \mathsf{B} + \mathsf{A}, S_\ell, \overline{B}_\ell, \overline{\gamma}^+_{k,l}, \overline{\gamma}^-_{k,\ell-1} \right) - \mathsf{A} \right)^2 \right]$$

$$\lim_{N_\ell\to\infty} \tfrac{1}{N_\ell} \left\| \widehat{\mathbf{z}}^+_{k\ell} - \mathbf{z}^0_\ell \right\|^2 = \tfrac{1}{N_\ell} \left\| \mathbf{V}^\top_\ell \left(\widehat{\mathbf{z}}^+_{k\ell} - \mathbf{z}^0_\ell \right) \right\|^2$$

$$\overset{a.s.}{=} \mathbb{E}\left[\left(G^+_\ell \left(\mathsf{C} + \mathsf{D}, \mathsf{B} + \mathsf{A}, S_\ell, \overline{B}_\ell, \overline{\gamma}^+_{kl}, \overline{\gamma}^-_{k,\ell-1} \right) - \mathsf{D} \right)^2 \right],$$

where $\mathsf{D} = S_\ell \mathsf{A} + \overline{B}_\ell + \overline{\Xi}_\ell$. Here we used the rotational invariance of the ℓ_2 norm, and the fact that equation (1a) is equivalent to $\mathbf{V}^\top_\ell \mathbf{z}^0_\ell = \operatorname{Diag}(\boldsymbol{s}_\ell) \mathbf{V}_{\ell-1} \mathbf{z}^0_{\ell-1} + \overline{\mathbf{b}}_\ell$ using the SVD (13) of the weight matrices $\mathbf{W}_\ell$.

At the heart of the proof lies a key insight: due to the randomness of the unitary matrices $\mathbf{V}_\ell$, the quantities $(\mathbf{z}^0_\ell, \boldsymbol{r}^-_{k\ell} - \mathbf{z}^0_\ell, \boldsymbol{r}^+_{k,\ell-1} - \mathbf{z}^0_{\ell-1})$ are asymptotically jointly Gaussian for even ℓ, with the asymptotic covariance matrix of $\{(z^0_{\ell-1,n}, r^+_{k,\ell-1,n} - z^0_{\ell-1,n}, r^-_{k\ell,n} - z^0_{\ell,n})\}$ given by $\begin{bmatrix} \mathbf{K}^+_{k\ell} & \mathbf{0} \\ \mathbf{0} & \tau^-_{k\ell} \end{bmatrix}$, where $\mathbf{K}_{k\ell} \in \mathbb{R}^{2\times 2}$ and $\tau^-_{k\ell}$ is a scalar. After establishing the asymptotic Gaussianity of $(\mathbf{z}^0_\ell, \boldsymbol{r}^-_{k\ell} - \mathbf{z}^0_\ell, \boldsymbol{r}^+_{k,\ell-1} - \mathbf{z}^0_{\ell-1})$, since $\widehat{\mathbf{z}}_\ell$ and $\widehat{\mathbf{z}}_{\ell-1}$ are componentwise functions of this triplet, we have the PL(2) convergence result in (31). Similarly, for odd ℓ, we can show that $(\mathbf{V}_{\ell-1} \mathbf{z}^0_{\ell-1}, \mathbf{V}_{\ell-1} \boldsymbol{r}^+_{k,\ell-1}, \mathbf{V}^\top_\ell \boldsymbol{r}^-_{k\ell})$ is asymptotically Gaussian. For these ℓ, $\mathbf{V}_{\ell-1} \widehat{\mathbf{z}}^-_{k,\ell-1}$ and $\mathbf{V}^\top_\ell \widehat{\mathbf{z}}^+_{k\ell}$ are functions of the triplet, which gives the result in (34).

Due to the asymptotic normality mentioned above, the inputs $(\boldsymbol{r}^-_\ell, \boldsymbol{r}^+_{\ell-1})$ to the estimators $\mathbf{g}^\pm_\ell$ are the true signals $(\mathbf{z}^0_{\ell-1}, \mathbf{z}^0_\ell)$ plus additive white Gaussian noise (AWGN). Hence, the estimators $\mathbf{g}^\pm_\ell$ act as denoisers, and ML-VAMP effectively reduces the inference problem 2 into a sequence of linear transformations and denoising problems. The denoising problems are solved by $\mathbf{g}^\pm_\ell$ for even ℓ, and by $\mathbf{G}^\pm_\ell$ for odd ℓ.

C. MMSE Estimation and Connections to the Replica Predictions

We next consider the special case of using MMSE estimators corresponding to the true distributions. In this case, the SE equations simplify considerably using the following *MSE functions*: let $\widehat{\mathbf{z}}^-_{\ell-1}$, $\widehat{\mathbf{z}}^+_\ell$ be the MMSE estimates of $\mathbf{z}^0_{\ell-1}$ and $\mathbf{z}^0_\ell$ from the variables $\mathbf{r}^+_{\ell-1}, \mathbf{r}^-_\ell$ under the joint density (10). Let $\mathcal{E}^\pm(\cdot)$ be the corresponding mean squared errors,

$$\mathcal{E}^+_\ell(\overline{\gamma}^+_{\ell-1}, \overline{\gamma}^-_\ell) := \lim_{N\to\infty} \frac{1}{N} \mathbb{E} \left\| \mathbf{z}^0_\ell - \widehat{\mathbf{z}}^+_\ell \right\|^2,$$

$$\mathcal{E}^-_{\ell-1}(\overline{\gamma}^+_{\ell-1}, \overline{\gamma}^-_\ell) := \lim_{N\to\infty} \frac{1}{N} \mathbb{E} \left\| \mathbf{z}^0_{\ell-1} - \widehat{\mathbf{z}}^-_{\ell-1} \right\|^2. \tag{36}$$

Theorem 4 (MSE of MMSE-ML-VAMP): Consider the system under the assumptions of Theorem 3, with MMSE estimation functions $\mathbf{g}^\pm_\ell, \mathbf{g}^+_0, \mathbf{g}^-_L$ from (11) for the belief estimates in (10) with $\gamma^+_{k\ell} = \overline{\gamma}^\pm_{k\ell}$ from the state-evolution equations. Then, the state evolution equations reduce to

$$\overline{\gamma}^+_{k\ell} = \frac{1}{\mathcal{E}^+_\ell\left(\overline{\gamma}^-_{k\ell}, \overline{\gamma}^+_{k,\ell-1} \right)} - \overline{\gamma}^-_{k\ell},$$

$$\overline{\gamma}^-_{k+1,\ell} = \frac{1}{\mathcal{E}^-_\ell\left(\overline{\gamma}^-_{k+1,\ell+1}, \overline{\gamma}^+_{k\ell} \right)} - \overline{\gamma}^+_{k\ell}, \tag{37}$$

where $1/\overline{\eta}^+_{k\ell} = \mathcal{E}^+_\ell(\overline{\gamma}^-_{k\ell}, \overline{\gamma}^+_{k,\ell-1})$ is the MSE of the estimate $\widehat{\mathbf{z}}^+_{k\ell}$.

Proof: See Appendix D. ■

Since the estimation functions in Theorem 4 are the MSE optimal functions for true densities, we will call this selection of estimation functions the *MMSE matched estimators*. Under the assumption of MMSE matched estimators, the theorem shows that the MSE error has a simple set of recursive expressions.

It is useful to compare the predicted MSE with the predicted optimal values. The works [48], [49] postulate the optimal MSE for inference in deep networks under the LSL model described above using the replica method from statistical physics. Interestingly, it is shown in [48, Th. 2] that the predicted minimum MSE satisfies equations that exactly agree with the fixed points of the updates (37). Thus, when the fixed points of (37) are unique, ML-VAMP with matched MMSE estimators provably achieves the Bayes optimal MSE predicted by the replica method. Although the replica method is not rigorous, this MSE predictions have been indepedently proven for the Gaussian case in [48] and certain two layer networks in [49]. This situation is similar to several other works relating the MSE of AMP with replica predictions [51], [52], [71]. The consequence is that, if the replica method is correct, ML-VAMP provides a computationally efficient method for inference with testable conditions under which it achieves the Bayes optimal MSE.

V. Numerical Simulations

We now numerically investigate the MAP-ML-VAMP and MMSE-ML-VAMP algorithms using two sets of experiments, where in each case the goal was to solve an estimation problem of the form in (2) using a neural network of the form in (1).

We used the Python 3.7 implementation of the ML-VAMP algorithm available on GitHub.[2]

The first set of experiments uses random draws of a synthetic network to validate the claims made about the ML-VAMP state-evolution (SE) in Theorem 3. In addition, it compares MAP-ML-VAMP and MMSE-ML-VAMP to the MAP approach (4) using a standard gradient-based solver, ADAM [72]. The second set of experiments applies ML-VAMP to image inpainting, using images of handwritten digits from the widely used MNIST dataset. Here, MAP-ML-VAMP and MMSE-ML-VAMP are respectively compared to the optimization approach (4) using the ADAM solver, and Stochastic Gradient Langevin Dynamics (SGLD) [28], an MCMC-based sampling method that approximates $\mathbb{E}[\mathbf{z}|\boldsymbol{y}]$.

A. *Performance on a Synthetic Network*

We first considered a 7-layer neural network of the form in (1). The first six layers, with dimensions $N_0 = 20$, $N_1 = N_2 = 100$, $N_3 = N_4 = 500$, $N_5 = N_6 = 784$, formed a (deterministic) deep generative prior driven by i.i.d. Gaussian $\mathbf{z}_0^0$. The matrices $\mathbf{W}_1, \mathbf{W}_3, \mathbf{W}_5$ and biases $\mathbf{b}_1, \mathbf{b}_3, \mathbf{b}_5$ were drawn i.i.d. Gaussian, and the activation functions ϕ_2, ϕ_4, ϕ_6 were ReLU. The mean of the bias vectors $\mathbf{b}_\ell$ was chosen so that a fixed fraction, ρ, of the linear outputs were positive, so that only the fraction ρ of the ReLU outputs were non-zero. Because this generative network is random rather than trained, we refer to it as "synthetic." The final layer, which takes the form $\mathbf{y} = \mathbf{A}\mathbf{z}_6^0 + \boldsymbol{\xi}_6$, generates noisy, compressed measurements of $\mathbf{z}_6^0$. Similar to [73], the matrix $\mathbf{A} \in \mathbb{R}^{M \times N_6}$ was constructed from the SVD $\mathbf{A} = \boldsymbol{U}\,\mathrm{Diag}(\boldsymbol{s})\mathbf{V}^{\mathsf{T}}$, where the singular-vector matrices $\boldsymbol{U}$ and $\mathbf{V}$ were drawn uniformly from the set of orthogonal matrices, and the singular values were geometrically spaced (i.e., $s_i/s_{i-1} = \kappa\ \forall i$) to achieve a condition number of $s_1/s_M = 10$. It is known that such matrices cause standard AMP algorithms to fail [73], but not VAMP algorithms [36]. The number of compressed measurements, M, was varied from 10 to 300, and the noise vector $\boldsymbol{\xi}$ was drawn i.i.d. Gaussian with a variance set to achieve a signal-to-noise ratio of $10\log_{10}(\mathbb{E}\|\mathbf{A}\mathbf{z}_6^0\|^2/\mathbb{E}\|\boldsymbol{\xi}\|^2) = 30$ dB.

To quantify the performance of ML-VAMP, we repeated the following 1000 times. First, we drew a random neural network as described above. Then we ran the ML-VAMP algorithm for 100 iterations, recording the normalized MSE (in dB) of the iteration-k estimate of the network input, $\widehat{\mathbf{z}}_{k0}^{\pm}$:

$$\mathrm{NMSE}\big(\widehat{\mathbf{z}}_{k0}^{\pm}\big) := 10\log_{10}\left[\frac{\big\|\mathbf{z}_0^0 - \widehat{\mathbf{z}}_{k0}^{\pm}\big\|^2}{\big\|\mathbf{z}_0^0\big\|^2}\right].$$

Since ML-VAMP computes two estimates of $\mathbf{z}_0^0$ at each iteration, we consider each estimate as corresponding to a "half iteration."

a) Validation of SE prediction: For MMSE-ML-VAMP, the left panel of Fig. 3 shows the NMSE versus half-iteration for $M = 100$ compressed measurements. The value shown is the average over 1000 random realizations. Also shown is the MSE predicted by the ML-VAMP state evolution. Comparing the two traces, we see that the SE predicts the actual behavior of MMSE-ML-VAMP remarkably well, within approximately 1 dB. The right panel shows the NMSE after $k = 50$ iterations (i.e., 100 half-iterations) for several number of measurements M. Again we see an excellent agreement between the actual MSE and the SE prediction. In both cases we used the positive fraction $\rho = 0.4$. Analogous results are shown for MAP-ML-VAMP in Fig. 4. There we see an excellent agreement between the actual MSE and the SE prediction for iterations $k \geq 15$ and all values of M.

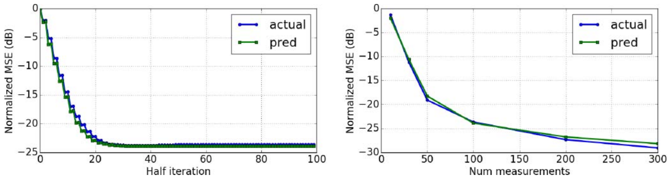

Fig. 3. NMSE of MMSE-ML-VAMP and its SE prediction when estimating the input to a randomly generated 7-layer neural network (see text of Section V-A). Left panel: Average NMSE versus half-iteration with $M = 100$ measurements. Right panel: Average NMSE verus measurements M after 50 iterations.

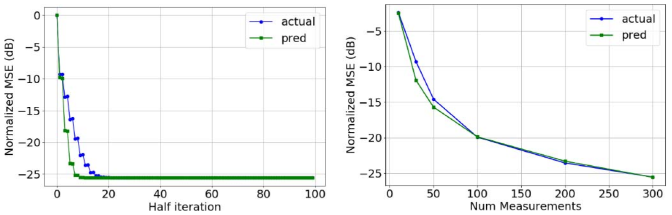

Fig. 4. Simulation with randomly generated neural network with MAP estimators from equation (12). Left panel: Normalized mean squared error (NMSE) for ML-VAMP and the predicted MSE as a function of the iteration with $M = 100$ measurements. Right panel: Final NMSE (50 iterations) for ML-VAMP and the predicted MSE as a function of the number of measurements, M. $\rho = 0.9$.

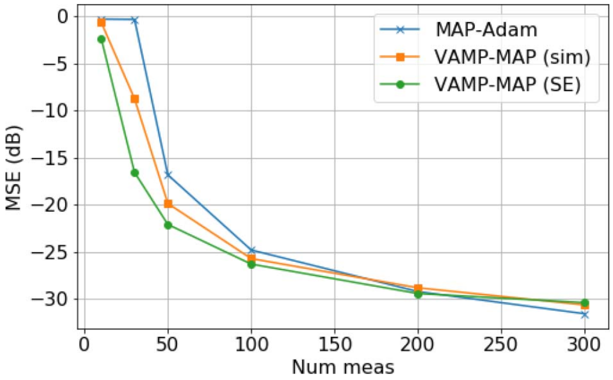

Fig. 5. Simulation with randomly generated neural network with MAP estimators from equation (12). Final NMSE for (a) MAP inference computed by Adam optimizer; (b) MAP inference from ML-VAMP; (c) State evolution prediction.

b) Comparison to ADAM: We now compare the MSE of MAP-ML-VAMP and its SE to that of the MAP approach (4) using the ADAM optimizer [72], as implemented in Tensorflow. As before, the goal was to recover the input $\mathbf{z}_0^0$ to the 7-layer synthetic network from a measurement of its output. Fig. 5 shows the median NMSE over 40 random network realizations for several values of M, the number of measurements. We see that, for $M \geq 100$,

[2] See https://github.com/GAMPTeam/vampyre.

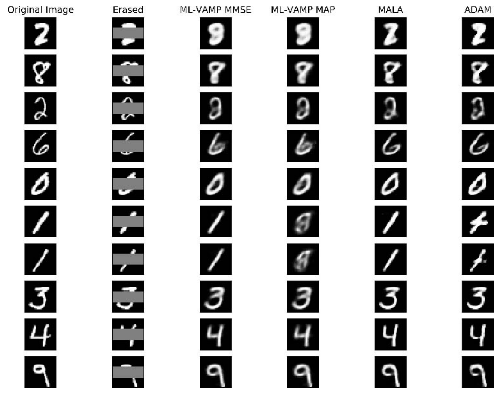

Fig. 6. MNIST inpainting: Original 28×28 images of handwritten digits (Col 1), with rows 10-20 are erased (Col 2). Comparison of reconstructions using MAP estimation with ADAM solver (Col 3), MAP estimation with ML-VAMP algorithm (Col 4), MMSE estimation with the Metropolis Adjusted Langevin Algorithm (Col 5), and MMSE estimation with ML-VAMP algorithm (Col 6).

the performance of MAP-ML-VAMP closely matches its SE prediction, as well as the performance of the ADAM-based MAP approach (4). For $M < 100$, there is a discrepancy between the MSE performance of MAP-ML-VAMP and its SE prediction, which is likely due to the relatively small dimensions involved. Also, for small M, MAP-ML-VAMP appears to achieve slightly better MSE performance than the ADAMP-based MAP approach (4). Since both are attempting to solve the same problem, the difference is likely due to ML-VAMP finding better local minima.

B. Image Inpainting: MNIST Dataset

To demonstrate that ML-VAMP can also work on a real-world dataset, we perform inpainting on the MNIST dataset. The MNIST dataset consists of $28 \times 28 = 784$ pixel images of handwritten digits, as shown in the first column of Fig. 6.

To start, we trained a 4-layer (deterministic) deep generative prior model from 50 000 digits using a variational autoencoder (VAE) [8]. The VAE "decoder" network was designed to accept 20-dimensional i.i.d. Gaussian random inputs $\mathbf{z}_0$ with zero mean and unit variance, and to produce MNIST-like images $\mathbf{x}$. In particular, this network began with a linear layer with 400 outputs, followed by a ReLU activations, followed by a linear layer with 784 units, followed by sigmoid activations that forced the final pixel values to between 0 and 1.

Given an image, $\mathbf{x}$, our measurement process produced $\mathbf{y}$ by erasing rows 10-20 of $\mathbf{x}$, as shown in the second column of Fig. 6. This process is known as "occlusion." By appending the occlusion layer onto our deep generative prior, we got a 5-layer network that generates an occluded MNIST image $\boldsymbol{y}$ from a random input $\mathbf{z}_0$. The "inpainting problem" is to recover the image $\boldsymbol{x} = \mathbf{z}_4$ from the occluded image $\boldsymbol{y}$.

For this inpainting problem, we compared MAP-ML-VAMP and MMSE-ML-VAMP to the MAP estimation approach (4) using the ADAM solver, and to Metropolis-Adjusted Langevin Algorithm (MALA) [28], [74], an MCMC-based sampling method that approximates $\mathbb{E}[\mathbf{z}|\boldsymbol{y}]$ by using discrete Langevin dynamics to generate proposal samples for Metropolis-Hastings algorithm [75]. Example image reconstructions are shown in Fig. 6. There we see that the qualitative performance of ML-VAMP is comparable to the baseline solvers.

VI. Conclusion

Inference using deep generative prior models provides a powerful tool for complex inverse problems. Rigorous theoretical analysis of these methods has been difficult due to the non-convex nature of the models. The ML-VAMP methodology for MMSE as well as MAP estimation provides a principled and computationally tractable method for performing the inference whose performance can be rigorously and precisely characterized in a certain large system limit. The approach thus offers a new and potentially powerful approach for understanding and improving deep neural network based models for inference.

References

[1] A. K. Fletcher, S. Rangan, and P. Schniter, "Inference in deep networks in high dimensions," in *Proc. IEEE Int. Symp. Inf. Theory*, 2018, pp. 1884–1888.

[2] P. Pandit, M. Sahraee, S. Rangan, and A. K. Fletcher, "Asymptotics of MAP inference in deep networks," in *Proc. IEEE Int. Symp. Inf. Theory*, 2019, pp. 842–846.

[3] R. Yeh, C. Chen, T. Y. Lim, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with perceptual and contextual losses," 2016. [Online]. Available: arXiv:1607.07539.

[4] A. Bora, A. Jalal, E. Price, and A. G. Dimakis, "Compressed sensing using generative models," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 537–546.

[5] Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*. Cambridge, U.K.: Cambridge Univ. Press, Jun. 2012.

[6] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal Stat. Soc. Ser. B, Methodol.*, vol. 58, no. 1, pp. 267–288, 1996.

[7] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2014, pp. 1278–1286.

[8] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013. [Online]. Available: arXiv:1312.6114.

[9] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015. [Online]. Available: arXiv:1511.06434.

[10] R. Salakhutdinov, "Learning deep generative models," *Annu. Rev. Stat. Appl.*, vol. 2, no. 1, pp. 361–385, 2015.

[11] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9446–9454.

[12] D. V. Veen, A. Jalal, M. Soltanolkotabi, E. Price, S. Vishwanath, and A. G. Dimakis, "Compressed sensing with deep image prior and learned regularization," 2018. [Online]. Available: arXiv:1806.06438.

[13] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. ACM Conf. Comput. Graph. Interact. Techn.*, 2000, pp. 417–424.

[14] P. McCullagh and J. A. Nelder, *Generalized Linear Models*, 2nd ed. Boca Raton, FL, USA: Chapman & Hall, 1989.

[15] A. Mousavi, A. B. Patel, and R. G. Baraniuk, "A deep learning approach to structured signal recovery," in *Proc. Allerton Conf. Commun. Control Comput.*, 2015, pp. 1336–1343.

[16] C. A. Metzler, A. Mousavi, and R. Baraniuk, "Learned D-AMP: Principled neural network based compressive image recovery," in *Proc. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 1772–1783.

[17] M. Borgerding, P. Schniter, and S. Rangan, "AMP-inspired deep networks for sparse linear inverse problems," *IEEE Trans. Signal Process.*, vol. 65, no. 16, pp. 4293–4308, Aug. 2017.

[18] J. H. R. Chang, C.-L. Li, B. Poczos, B. V. K. V. Kumar, and A. C. Sankaranarayanan, "One network to solve them all—Solving linear inverse problems using deep projection models," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5889–5898.

[19] P. Hand and V. Voroninski, "Global guarantees for enforcing deep generative priors by empirical risk," 2017. [Online]. Available: arXiv:1705.07576.

[20] M. Kabkab, P. Samangouei, and R. Chellappa, "Task-aware compressed sensing with generative adversarial networks," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 2297–2304.

[21] V. Shah and C. Hegde, "Solving linear inverse problems using GAN priors: An algorithm with provable guarantees," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 4609–4613.

[22] S. Tripathi, Z. C. Lipton, and T. Q. Nguyen, "Correction by projection: Denoising images with generative adversarial networks," 2018. [Online]. Available: arXiv:1803.04477.

[23] D. G. Mixon and S. Villar, "Sunlayer: Stable denoising with generative networks," 2018. [Online]. Available: arXiv:1803.09319.

[24] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5188–5196.

[25] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," 2015. [Online]. Available: arXiv:1506.06579.

[26] V. Dumoulin *et al.*, "Adversarially learned inference," 2016. [Online]. Available: arXiv:1606.00704.

[27] X. Cheng, N. S. Chatterji, Y. Abbasi-Yadkori, P. L. Bartlett, and M. I. Jordan, "Sharp convergence rates for Langevin dynamics in the nonconvex setting," 2018. [Online]. Available: arXiv:1805.01648.

[28] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient Langevin dynamics," in *Proc. 28th Int. Conf. Mach.Learn.*, 2011, pp. 681–688.

[29] 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.

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

[31] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," in *Proc. IEEE Int. Symp. Inf. Theory*, 2011, pp. 2168–2172.

[32] A. K. Fletcher and P. Schniter, "Learning and free energies for vector approximate message passing," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2017, pp. 4247–4251.

[33] A. K. Fletcher, P. Pandit, S. Rangan, S. Sarkar, and P. Schniter, "Plug-in estimation in high-dimensional linear inverse problems: A rigorous analysis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7440–7449.

[34] S. Sarkar, A. K. Fletcher, S. Rangan, and P. Schniter, "Bilinear recovery using adaptive vector-AMP," *IEEE Trans. Signal Process.*, vol. 67, no. 13, pp. 3383–3396, Jul. 2019.

[35] J. Barbier, N. Macris, M. Dia, and F. Krzakala, "Mutual information and optimality of approximate message-passing in random linear estimation," 2017. [Online]. Available: arXiv:1701.05823.

[36] S. Rangan, P. Schniter, and A. K. Fletcher, "Vector approximate message passing," *IEEE Trans. Inf. Theory*, vol. 65, no. 10, pp. 6664–6684, Oct. 2019.

[37] T. P. Minka, "Expectation propagation for approximate Bayesian inference," in *Proc. Conf. Uncertainty Artif. Intell. (UAI)*, 2001, pp. 362–369.

[38] K. Takeuchi, "Rigorous dynamics of expectation-propagation-based signal recovery from unitarily invariant measurements," in *Proc. IEEE Int. Symp. Inf. Theory*, 2017, pp. 501–505.

[39] M. Opper and O. Winther, "Expectation consistent approximate inference," *J. Mach. Learn. Res.*, vol. 6, pp. 2177–2204, Dec. 2005.

[40] A. K. Fletcher, M. Sahraee-Ardakan, S. Rangan, and P. Schniter, "Expectation consistent approximate inference: Generalizations and convergence," in *Proc. IEEE Int. Symp. Inf. Theory*, 2016, pp. 190–194.

[41] B. Çakmak, O. Winther, and B. H. Fleury, "S-AMP: Approximate message passing for general matrix ensembles," in *Proc. IEEE Inf. Theory Workshop (ITW)*, 2014, pp. 192–196.

[42] J. Ma and L. Ping, "Orthogonal AMP," *IEEE Access*, vol. 5, pp. 2020–2033, 2017.

[43] A. Manoel, F. Krzakala, M. Mézard, and L. Zdeborová, "Multi-layer generalized linear estimation," in *Proc. IEEE Int. Symp. Inf. Theory*, 2017, pp. 2098–2102.

[44] F. Krzakala, A. Manoel, E. W. Tramel, and L. Zdeborová, "Variational free energies for compressed sensing," in *Proc. IEEE Int. Symp. Inf. Theory*, 2014, pp. 1499–1503.

[45] S. Rangan, P. Schniter, E. Riegler, A. K. Fletcher, and V. Cevher, "Fixed points of generalized approximate message passing with arbitrary matrices," *IEEE Trans. Inf. Theory*, vol. 62, no. 12, pp. 7464–7474, Dec. 2016.

[46] 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.

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

[48] G. Reeves, "Additivity of information in multilayer networks via additive Gaussian noise transforms," in *Proc. Allerton Conf. Commun. Control Comput.*, 2017, pp. 1064–1070.

[49] M. Gabrié *et al.*, "Entropy and mutual information in models of deep neural networks," in *Proc. Conf. Neural Inf. Process. Syst. (NIPS)*, 2018, pp. 1826–1836.

[50] J. Barbier, F. Krzakala, N. Macris, L. Miolane, and L. Zdeborová, "Optimal errors and phase transitions in high-dimensional generalized linear models," *Proc. Nat. Acad. Sci.*, vol. 116, no. 12, pp. 5451–5460, 2019.

[51] G. Reeves and H. D. Pfister, "The replica-symmetric prediction for compressed sensing with Gaussian matrices is exact," in *Proc. IEEE Int. Symp. Inf. Theory*, 2016, pp. 665–669.

[52] J. Barbier, M. Dia, N. Macris, and F. Krzakala, "The mutual information in random linear estimation," in *Proc. 54th Annu. Allerton Conf. Commun. Control Comput. (Allerton)*, 2016, pp. 625–632.

[53] R. M. Neal, *Bayesian Learning for Neural Networks*, vol. 118. New York, NY, USA: Springer-Verlag, 1996. [Online]. Available: https://link.springer.com/content/pdf/10.1007%2F978-1-4612-0745-0.pdf

[54] R. Giryes, G. Sapiro, and A. M. Bronstein, "Deep neural networks with random Gaussian weights: A universal classification strategy?" *IEEE Trans. Signal Process.*, vol. 64, no. 13, pp. 3444–3457, Jul. 2016.

[55] B. Hanin and D. Rolnick, "How to start training: The effect of initialization and architecture," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 571–581.

[56] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun, "The loss surfaces of multilayer networks," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2015, pp. 192–204.

[57] P. Li and P.-M. Nguyen, "On random deep weight-tied autoencoders: Exact asymptotic analysis, phase transitions, and implications to training," in *Proc. Int. Conf. Learn. Res. (ICLR)*, 2019. [Online]. Available: https://openreview.net/forum?id=HJx54i05tX

[58] S. S. Schoenholz, J. Gilmer, S. Ganguli, and J. Sohl-Dickstein, "Deep information propagation," 2016. [Online]. Available: arXiv:1611.01232.

[59] R. Novak *et al.*, "Bayesian deep convolutional networks with many channels are Gaussian processes," 2018. [Online]. Available: arXiv:1810.05148.

[60] W. Huang, P. Hand, R. Heckel, and V. Voroninski, "A provably convergent scheme for compressive sensing under random generative priors," 2018. [Online]. Available: arXiv:1812.04176.

[61] Q. Lei, A. Jalal, I. S. Dhillon, and A. G. Dimakis, "Inverting deep generative models, one layer at a time," 2019. [Online]. Available: arXiv:1906.07437.

[62] C. Rush and R. Venkataramanan, "Finite-sample analysis of approximate message passing algorithms," *IEEE Trans. Inf. Theory*, vol. 64, no. 11, pp. 7264–7286, Nov. 2018.

[63] 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.

[64] P. J. Huber and E. M. Ronchetti, *Robust Statistics*, 2nd ed. Hoboken, NJ, USA: Wiley, Feb. 2009. [Online]. Available: https://www.wiley.com/en-us/Robust+Statistics%2C+2nd+Edition-p-9780470129906

[65] P. Schniter, S. Rangan, and A. K. Fletcher, "Vector approximate message passing for the generalized linear model," in *Proc. Asilomar Conf. Signals Syst. Comput.*, 2016, pp. 1525–1529.

[66] A. K. Fletcher, S. Rangan, and P. Schniter, "Inference in deep networks in high dimensions," 2017. [Online]. Available: arXiv:1706.06549.

[67] B. He, H. Liu, J. Lu, and X. Yuan, "Application of the strictly contractive peaceman-rachford splitting method to multi-block separable convex programming," in *Splitting Methods in Communication, Imaging, Science, and Engineering* (Scientific Computation), R. Glowinski, S. Osher, and W. Yin, Eds. Cham, Switzerland: Springer, 2016. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-41589-5_6

[68] B. He, H. Liu, Z. Wang, and X. Yuan, "A strictly contractive Peaceman–Rachford splitting method for convex programming," *SIAM J. Optim.*, vol. 24, no. 3, pp. 1011–1040, 2014.

[69] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2282–2312, Jul. 2005.

[70] M. Pereyra *et al.*, "A survey of stochastic simulation and optimization methods in signal processing," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 2, pp. 224–241, Mar. 2016.

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

[72] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: arXiv:1412.6980.

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

[74] R. M. Neal *et al.*, "MCMC using hamiltonian dynamics," in *Handbook of Markov Chain Monte Carlo*, vol. 2. Boca Raton, FL, USA: CRC Press, 2011, p. 2.

[75] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, Apr. 1970. [Online]. Available: https://academic.oup.com/biomet/article/57/1/97/284580