Large-Scale Bayesian Tensor Reconstruction via Approximate Message Passing

Bingyang Cheng, Zhongtao Chen, Yichen Jin, Hao Zhang, Chen Zhang, Edmund Y. Lam*, and Yik-Chung Wu*
Department of Electrical and Computer Engineering
The University of Hong Kong
Hong Kong, China
*Corresponding authors: Edmund Y. Lam (elam@eee.hku.hk) and Yik-Chung Wu (ycwu@eee.hku.hk)

arXiv:2505.16305v3 [cs.LG] 29 Aug 2026

***Abstract*—While CANDECOMP/PARAFAC (CP) decomposition (CPD) is fundamental for tensor reconstruction, Bayesian CPD often scales poorly because variational updates require repeated matrix inversions. We develop CP generalized approximate message passing (CP-GAMP) for incomplete noisy Bayesian CPD. The algorithm uses Gaussian message approximations to avoid high-dimensional inversions, and it combines a Bernoulli-Gaussian prior with expectation-maximization updates to estimate effective CP rank and noise variance. We also give a formal state evolution (SE) recursion and relate its fixed points to replica-symmetric saddle points, so CP-GAMP's SE-predicted error can be compared with the formal replica-symmetric minimum mean-squared error (MMSE) benchmark in the matched limit. Synthetic and image-inpainting experiments show that CP-GAMP substantially reduces runtime relative to variational Bayesian CPD while maintaining competitive reconstruction accuracy. Source code is available at https://github.com/BingyangCheng/CP-GAMP.git.**

***Index Terms*—Tensor decomposition, tensor completion, approximate message passing, Bayesian inference, state evolution.**

I. Introduction

Tensors provide a natural representation for multiway data in data mining and machine learning, including user-item-time interactions, relational events, spatio-temporal measurements, hyperspectral images, and scientific recordings [1]–[4]. In these applications, the goal is often not only to fill in missing entries, but also to discover a compact set of latent factors that explain how different modes interact. CANDECOMP/PARAFAC (CP) decomposition (CPD) is especially appealing because it expresses a tensor as a sum of rank-one components with factors tied to the original modes [5], [6]. This interpretability makes CPD a useful model for exploratory data analysis, recommendation, anomaly detection, and signal reconstruction.

For real data, observations are typically noisy, heavily incomplete, and collected under an unknown effective rank. Classical optimization methods such as alternating least squares [6] can be efficient, but they usually require the rank to be chosen externally and do not provide posterior uncertainty. Bayesian CPD addresses these issues by placing priors on factor matrices and inferring latent factors, noise levels, and rank-related hyperparameters within one probabilistic model [7]–[9]. The difficulty is computational. Sampling-based Bayesian tensor methods are often too slow for large tensors [10], while variational inference (VI)-based Bayesian CPD methods repeatedly update covariance terms and perform matrix inversions whose cost grows rapidly with the rank and the mode dimensions [9], [11]–[14]. As a result, scalable Bayesian inference for large incomplete CP tensors remains challenging.

Approximate message passing (AMP) offers a different route to scalable Bayesian inference. AMP and generalized AMP (GAMP) replace high-dimensional posterior computations by scalar denoising steps plus residual corrections, yielding efficient algorithms with state evolution characterizations in several high-dimensional models [15]–[17]. Bilinear GAMP (Bi-GAMP) extends this principle to matrix factorization [18], [19], and tensor completion AMP (TC-AMP) applies bilinear inference after unfolding a tensor into matrices [20]. Nevertheless, unfolding can discard the component-sharing structure that distinguishes CPD from independent matrix factorizations. Existing tensor AMP analyses also tend to focus on idealized tensor estimation models, known priors, or fixed hyperparameters, which limits their direct use in incomplete real-data settings [21]. These limitations motivate an AMP-style Bayesian CPD method that operates directly on the CP factor graph, preserves the multilinear structure, and estimates effective rank and noise parameters.

This paper develops CP generalized approximate message passing (CP-GAMP), a scalable AMP algorithm for Bayesian CP tensor reconstruction from incomplete noisy observations. Starting from the CPD factor graph, we approximate the high-dimensional loopy belief-propagation messages by Gaussian messages and retain Onsager-style correction terms that compensate for self-feedback in the dense graph. The resulting updates avoid the matrix inversions used by variational Bayesian CPD and reduce inference to multilinear aggregation, scalar output updates, and scalar factor denoising. To make the method adaptive, we combine CP-GAMP with a Bernoulli-Gaussian (BG) prior and expectation-maximization (EM) updates [22], [23] so that inactive components can be pruned and the noise variance can be estimated from the data.

The main contributions are as follows. First, we derive CP-GAMP as a direct AMP approximation on the CP factor graph, rather than applying matrix factorization after tensor unfolding. Second, we introduce an adaptive CP-GAMP variant that uses a BG prior and EM updates to estimate component

© 2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

activity probabilities and noise variance, supporting data-driven effective-rank selection. Third, we provide a formal state evolution (SE) recursion for the matched Gaussian-prior large-system limit and relate its informative fixed points to replica-symmetric saddle point equations under explicit assumptions. Finally, we evaluate CP-GAMP on synthetic tensor reconstruction and image inpainting, including ablations on prior mismatch and non-Gaussian noise.

II. Related Work

Classical tensor decomposition methods such as CPD and Tucker decomposition generalize matrix factorization to multiway data [5], [6], [24]. Probabilistic CPD models handle missing entries and uncertainty [7], [8], while Bayesian CP factorization and nonnegative tensor completion can also determine the rank automatically [9], [25]. Their sampling or variational implementations remain costly at scale because of repeated posterior computations, including covariance updates and matrix inversions.

AMP was originally developed for compressed sensing and high-dimensional linear regression [15], [17]. GAMP handles separable nonlinear output channels [16], while Bi-GAMP addresses bilinear matrix factorization [18], [19]. TC-AMP applies AMP to tensor completion by unfolding a tensor into matrices [20], but this transformation does not directly enforce CP component consistency across all modes. The CP-GAMP algorithm developed here keeps the CP factor graph and introduces rank and noise learning in the message passing loop.

Statistical physics and information theory have also studied tensor estimation through AMP, state evolution, and replica arguments [26]–[28]. These works provide asymptotic thresholds and Bayes-optimal predictions for idealized rank-one tensor CPD models. In contrast, our CP-GAMP construction is written for CPD with an arbitrary working rank R, so the message updates and SE discussion retain the multilinear coupling among multiple rank-one components. The analysis follows the same tradition but is framed carefully: the Gaussian message approximation and the replica-symmetric characterization are modeling assumptions for the large-system analysis.

Nonparametric iterative machine teaching casts example selection as functional optimization [29], with related methods accelerating implicit neural representations, graph property learners, and attention learners [30]–[32]. These works reduce neural-network training cost; CP-GAMP instead reduces Bayesian tensor-inference cost on a multilinear factor graph.

III. Bayesian CPD Model

For an N-th order tensor $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, the CPD model used in [9] is

$$\boldsymbol{\mathcal{Z}} = \sum_{r=1}^{R} \boldsymbol{a}_{\cdot r}^{(1)} \circ \cdots \circ \boldsymbol{a}_{\cdot r}^{(N)} = [\![\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}]\!], \tag{1}$$

where $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$ is the mode-n factor matrix, $\boldsymbol{a}_{\cdot r}^{(n)}$ is its r-th column, $\circ$ is the vector outer product, and the double brackets denote the CP/Kruskal composition rather than concatenation. The entry indexed by $\boldsymbol{i} = (i_1, \ldots, i_N)$ is

$$z_{\boldsymbol{i}} = \sum_{r=1}^{R} \prod_{n=1}^{N} a_{i_n, r}^{(n)}. \tag{2}$$

We observe a noisy and incomplete tensor

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{O}} \odot (\boldsymbol{\mathcal{Z}} + \boldsymbol{\mathcal{W}}), \tag{3}$$

where $\boldsymbol{\mathcal{O}}$ is a binary observation mask with entries $o_{\boldsymbol{i}}$, $\Omega = \{\boldsymbol{i} : o_{\boldsymbol{i}} = 1\}$ is the observed index set, $\odot$ is elementwise multiplication, and $w_{\boldsymbol{i}} \sim \mathcal{N}(0, v^w)$ is additive white Gaussian noise (AWGN). For $\boldsymbol{i} \in \Omega$, $p(y_{\boldsymbol{i}} | z_{\boldsymbol{i}}) = \mathcal{N}(y_{\boldsymbol{i}}; z_{\boldsymbol{i}}, v^w)$; for $\boldsymbol{i} \notin \Omega$, the likelihood carries no information about $z_{\boldsymbol{i}}$.

To learn the effective CP rank, we place a BG prior on each factor entry:

$$p(a_{i_n, r}^{(n)}) = (1 - \lambda_r)\delta(a_{i_n, r}^{(n)}) + \lambda_r \mathcal{N}(a_{i_n, r}^{(n)}; 0, 1). \tag{4}$$

The shared activity parameter λ_r controls whether component r is active across all modes. The posterior is proportional to the separable likelihood times the product of these factor priors:

$$p(\{\mathbf{A}^{(n)}\}_{n=1}^{N} | \boldsymbol{\mathcal{Y}}) \propto \prod_{\boldsymbol{i} \in \Omega} p\left(y_{\boldsymbol{i}} \middle| \sum_{r=1}^{R} \prod_{n=1}^{N} a_{i_n, r}^{(n)} \right) \prod_{n, i_n, r} p(a_{i_n, r}^{(n)}). \tag{5}$$

IV. Method

A. Factor Graph and Message Approximation

The factor graph contains one likelihood node for each observed tensor entry and one variable node for each factor entry. Fig. 1(a) illustrates the complete CP factor graph. The graph is dense and loopy: for instance, two observations that share mode indices are connected to common factor variables, creating short cycles through likelihood nodes and shared variable nodes, as shown in Fig. 1(b). Exact loopy belief propagation is therefore computationally infeasible because factor-to-variable messages require integrals over all neighboring factor entries [33]–[35].

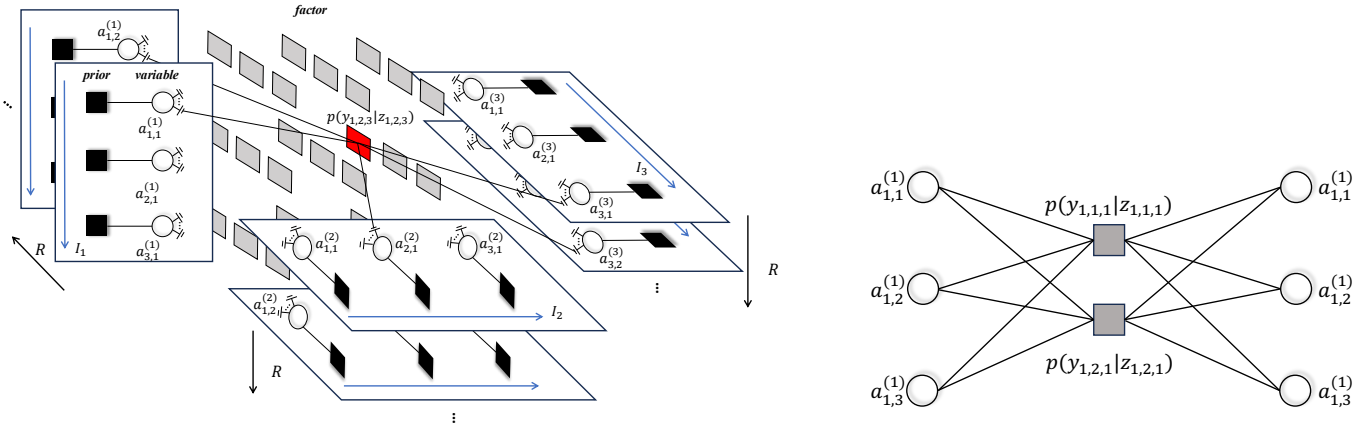
Assumption 1 (Gaussian message approximation): In the large-system regime where $I_1, \ldots, I_N, R$ grow with fixed ratios I_n/R, the aggregate messages entering each scalar denoiser can be approximated by Gaussian random variables whose means and variances are tracked by scalar state variables.

Assumption 1 is the standard AMP-style approximation used to obtain tractable updates and state-evolution characterizations [15]–[17]. Under this approximation, CP-GAMP reduces the high-dimensional message updates to scalar posterior mean and variance computations.

B. Algorithm

Let $\hat{a}_{i_n, r}^{(n)}(t)$ and $v_{i_n, r}^{a^{(n)}}(t)$ denote the approximate posterior mean and variance at iteration t. For each tensor entry, CP-GAMP forms the predicted mean and variance

$$\hat{p}_{\boldsymbol{i}}(t) = \sum_{r=1}^{R} \prod_{n=1}^{N} \hat{a}_{i_n, r}^{(n)}(t), \tag{6}$$



(a) Complete CP factor graph

(b) Local short-loop structure

Fig. 1. Factor-graph view of Bayesian CPD. Black squares are prior factors, white circles are factor variables, and gray or red squares are likelihood factors. The plates span mode sizes I_n and component index R. Panel (b) isolates two likelihood nodes sharing variables; unshared neighbors are omitted.

$$v_{\boldsymbol{i}}^{p}(t)=\sum_{r=1}^{R}\left[\prod_{n}\left(v_{i_n,r}^{a^{(n)}}(t)+(\hat{a}_{i_n,r}^{(n)}(t))^2\right)-\prod_{n}(\hat{a}_{i_n,r}^{(n)}(t))^2\right]. \tag{7}$$

Combining this Gaussian prediction with the likelihood gives the approximate posterior moments $\hat{z}_{\boldsymbol{i}}(t)$ and $v_{\boldsymbol{i}}^{z}(t)$. For the AWGN missing-data channel,

$$\hat{z}_{\boldsymbol{i}}(t)=\begin{cases}\dfrac{y_{\boldsymbol{i}}v_{\boldsymbol{i}}^{p}(t)+\hat{p}_{\boldsymbol{i}}(t)v^{w}}{v_{\boldsymbol{i}}^{p}(t)+v^{w}}, & o_{\boldsymbol{i}}=1,\\ \hat{p}_{\boldsymbol{i}}(t), & o_{\boldsymbol{i}}=0,\end{cases} \tag{8}$$

$$v_{\boldsymbol{i}}^{z}(t)=\begin{cases}\left(1/v_{\boldsymbol{i}}^{p}(t)+1/v^{w}\right)^{-1}, & o_{\boldsymbol{i}}=1,\\ v_{\boldsymbol{i}}^{p}(t), & o_{\boldsymbol{i}}=0.\end{cases} \tag{9}$$

The scaled residual and inverse residual variance are

$$\hat{s}_{\boldsymbol{i}}(t)=\frac{\hat{z}_{\boldsymbol{i}}(t)-\hat{p}_{\boldsymbol{i}}(t)}{v_{\boldsymbol{i}}^{p}(t)},\quad v_{\boldsymbol{i}}^{s}(t)=\frac{1}{v_{\boldsymbol{i}}^{p}(t)}\left(1-\frac{v_{\boldsymbol{i}}^{z}(t)}{v_{\boldsymbol{i}}^{p}(t)}\right). \tag{10}$$

For a factor variable $a_{i_n,r}^{(n)}$, let $\mathcal{F}_{i_n,r}^{(n)}$ denote the set of observed entries adjacent to it, write $\mathcal{F}=\mathcal{F}_{i_n,r}^{(n)}$ in the following equations, and define $\hat{a}_{\boldsymbol{i},r}^{(\backslash n)}(t)=\prod_{\ell\neq n}\hat{a}_{i_\ell,r}^{(\ell)}(t)$. The effective scalar observation variance and mean are

$$v_{i_n,r}^{q}(t)=\left(\sum_{\boldsymbol{i}'\in\mathcal{F}}\left(\hat{a}_{\boldsymbol{i}',r}^{(\backslash n)}(t)\right)^2 v_{\boldsymbol{i}'}^{s}(t)\right)^{-1}, \tag{11}$$

$$\hat{q}_{i_n,r}(t)=\hat{a}_{i_n,r}^{(n)}(t)+v_{i_n,r}^{q}(t)\sum_{\boldsymbol{i}'\in\mathcal{F}}\hat{a}_{\boldsymbol{i}',r}^{(\backslash n)}(t)\hat{s}_{\boldsymbol{i}'}(t)+\Delta_{\mathrm{Ons}}, \tag{12}$$

where the Onsager correction is

$$\Delta_{\mathrm{Ons}}=\hat{a}_{i_n,r}^{(n)}(t)v_{i_n,r}^{q}(t)\sum_{\boldsymbol{i}'\in\mathcal{F}}\omega_{\boldsymbol{i}',r}^{(\backslash n)}(t)\left((\hat{s}_{\boldsymbol{i}'}(t))^2-v_{\boldsymbol{i}'}^{s}(t)\right), \tag{13}$$

with

$$\omega_{\boldsymbol{i},r}^{(\backslash n)}(t)=\prod_{\ell\neq n}\left(v_{i_\ell,r}^{a^{(\ell)}}(t)+(\hat{a}_{i_\ell,r}^{(\ell)}(t))^2\right)-\prod_{\ell\neq n}(\hat{a}_{i_\ell,r}^{(\ell)}(t))^2. \tag{14}$$

This term comes from the first-order variance perturbation retained in the AMP expansion [16], [18], [36]. The denoiser then computes

$$\begin{aligned}\hat{a}_{i_n,r}^{(n)}(t+1)&=g(\hat{q}_{i_n,r}(t),v_{i_n,r}^{q}(t)),\\ v_{i_n,r}^{a^{(n)}}(t+1)&=v_{i_n,r}^{q}(t)g'(\hat{q}_{i_n,r}(t),v_{i_n,r}^{q}(t)).\end{aligned} \tag{15}$$

For the BG prior, g is the posterior mean under a scalar Gaussian observation and has a closed form. The posterior activity probability is

$$\pi(\hat{q},v^q;\lambda_r)=\left[1+\left(\frac{\lambda_r}{1-\lambda_r}\frac{\mathcal{N}(\hat{q};0,1+v^q)}{\mathcal{N}(\hat{q};0,v^q)}\right)^{-1}\right]^{-1}. \tag{16}$$

Given this activity probability, the BG denoiser is

$$g(\hat{q},v^q)=\pi(\hat{q},v^q;\lambda_r)\gamma(\hat{q},v^q), \tag{17}$$

$$\gamma(\hat{q},v^q)=\frac{\hat{q}/v^q}{1+1/v^q},\quad \nu(\hat{q},v^q)=\frac{1}{1+1/v^q}. \tag{18}$$

The corresponding posterior variance is

$$v^q g'(\hat{q},v^q)=\pi(\nu+|\gamma|^2)-\pi^2|\gamma|^2, \tag{19}$$

where π, γ, and ν use the same arguments. Equations (17)-(19) make the rank-learning mechanism transparent: when the scalar observation for a component is not sufficiently supported by the data, π decreases, the posterior mean shrinks toward zero, and EM subsequently reduces λ_r.

C. EM Hyperparameter Learning

The EM update for the BG activity probability is

$$\lambda_r^{j+1}=\frac{1}{\sum_{n=1}^{N}I_n}\sum_{n=1}^{N}\sum_{i_n=1}^{I_n}\pi(\hat{q}_{i_n,r},v_{i_n,r}^{q};\lambda_r^{j}). \tag{20}$$

The noise variance update is

$$v^{w,j+1}=\frac{1}{|\Omega|}\sum_{\boldsymbol{i}\in\Omega}\left(|y_{\boldsymbol{i}}-\hat{z}_{\boldsymbol{i}}|^2+v_{\boldsymbol{i}}^{z}\right). \tag{21}$$

Algorithm 1 Adaptive CP-GAMP

Require: $\boldsymbol{\mathcal{Y}}$, mask $\boldsymbol{\mathcal{O}}$, maximum rank R, maximum iterations T, pruning threshold T_λ

Initialize $\hat{a}_{i_n,r}^{(n)}$, $v_{i_n,r}^{a^{(n)}}$, λ_r, and v^w
for $t = 1$ to T **do**
  Update $\hat{p}_{\boldsymbol{i}}$, $v_{\boldsymbol{i}}^p$ using (6)-(7)
  Update $\hat{z}_{\boldsymbol{i}}$, $v_{\boldsymbol{i}}^z$, $\hat{s}_{\boldsymbol{i}}$, $v_{\boldsymbol{i}}^s$ using (8)-(10)
  Update $\hat{q}_{i_n,r}$, $v_{i_n,r}^q$ using (11)-(12)
  Denoise each factor entry using (15)
  Update λ_r and v^w using (20)-(21)
  Remove component r if $\lambda_r < T_\lambda$
end for
return factor estimates and reconstructed tensor $\hat{\boldsymbol{\mathcal{Z}}}$

The pruning step in Algorithm 1 is used after the EM update to convert the over-parameterized model into a parsimonious CP representation. It is a practical model selection step.

D. Computational Complexity and Implementation

For a dense tensor, the dominant computation in each CP-GAMP iteration is the sweep over tensor entries used to update $\hat{p}_{\boldsymbol{i}}$, $v_{\boldsymbol{i}}^p$, $\hat{s}_{\boldsymbol{i}}$, and $v_{\boldsymbol{i}}^s$. For an incomplete tensor, this sweep can be restricted to observed entries, while scalar denoising and BG activity updates scan the factor entries. The per-iteration complexity is therefore

$$O\left(NR|\Omega| + R\sum_{n=1}^{N} I_n\right), \tag{22}$$

where Ω is the observed index set. The first term is the multilinear prediction and residual aggregation over observations, and the second term is the factor-side scalar denoising and activity update cost. The noise-variance update costs $O(|\Omega|)$ and is lower order. In contrast, VI-based Bayesian CPD methods perform mode-wise covariance updates whose cost grows cubically in the rank in the matrix-inversion step and also requires repeated Khatri-Rao products [9]. The advantage of CP-GAMP is most visible when $|\Omega|$ is large but the target rank is moderate or over-specified.

The update equations are separable over tensor entries and over factor rows after the residuals are computed. This gives a direct parallel implementation: entry-wise updates can be distributed across observed indices, while denoising can be performed independently for every factor entry.

V. State Evolution and Replica-Symmetric Characterization

We now summarize the formal asymptotic approximation used to interpret CP-GAMP performance. The analysis is based on the following assumptions.

Assumption 2 (Large-system limit): $I_1, \ldots, I_N, R \to \infty$ with fixed ratios $\alpha_n = I_n/R$.

Assumption 3 (Matched generative model): Factor entries are generated independently from the prior used by the denoiser, the observation mask is independent and identically distributed (i.i.d.) Bernoulli with probability ρ, and the noise is i.i.d. Gaussian with variance v^w.

Let $\chi_a^{(n)} = \mathbb{E}[(a^{(n)})^2]$ and let $q_a^{(n)}(t)$ denote the estimate second moment tracked by SE. Define $\chi_z = R\prod_{n=1}^{N}\chi_a^{(n)}$. Starting from a weakly informative initialization $q_a^{(n)}(0) = \rho_0\chi_a^{(n)}$ with a small $0 < \rho_0 < 1$, the SE recursion is

$$V(t) = \chi_z - R\prod_{n=1}^{N} q_a^{(n)}(t), \tag{23}$$

$$\Sigma_a^{(n)}(t+1) = \frac{v^w + V(t)}{\rho\prod_{\ell\neq n} I_\ell q_a^{(\ell)}(t)}, \tag{24}$$

$$q_a^{(n)}(t+1) = \mathbb{E}\left[\left(\mathbb{E}[a^{(n)}|a^{(n)} + \sqrt{\Sigma_a^{(n)}(t+1)}\xi]\right)^2\right], \tag{25}$$

$$V(t+1) = \chi_z - R\prod_{n=1}^{N} q_a^{(n)}(t+1), \tag{26}$$

where $\xi \sim \mathcal{N}(0,1)$. For a Gaussian prior with variance $\chi_a^{(n)}$, (25) reduces to

$$q_a^{(n)}(t+1) = \frac{(\chi_a^{(n)})^2}{\chi_a^{(n)} + \Sigma_a^{(n)}(t+1)}. \tag{27}$$

Proposition 1 (Conditional correspondence): Under the large-system, matched-prior, and replica-symmetry assumptions, and when CP-GAMP converges to the informative fixed point, the formal fixed-point equations of (23)-(26) match the replica-symmetric saddle point equations for the corresponding CPD model.

A derivation of this SE-replica equivalence, following the SE and replica strategy of multi-layer Bi-GAMP (ML-BiGAMP) [36] and the classical replica method [37] but adapted to the CP tensor factor graph, is given in Appendix D. This also connects the replica calculation to the Bayes benchmark in tensor estimation [27], [28]: at the replica-symmetric optimum, the tensor-error order parameter $V^\star$ predicts the normalized minimum mean-squared error (MMSE) $V^\star/\chi_z$. SE tracks the branch reached by CP-GAMP from its initialization. It matches this formal MMSE benchmark only when the stable informative fixed point coincides with the replica-symmetric optimum, while coexistence of branches indicates a possible algorithmic gap. For $N = 2$, the CP model reduces to a bilinear factorization of the ML-BiGAMP type, but arbitrary order requires tracking mode-wise overlaps and multilinear Onsager terms, so the lift is not automatic. This is a formal correspondence rather than a finite-sample convergence result for adaptive BG-EM. The SE comparison below uses matched Gaussian-prior CP-GAMP without EM hyperparameter learning or pruning.

VI. Experiments

A. Experiment Setup

Synthetic tensors are generated from (1) with factor entries drawn from $\mathcal{N}(0,1)$. AWGN is added at a target signal-to-noise ratio (SNR), and entries are independently sampled

TABLE I
IMAGE INPAINTING RESULTS AT OBSERVATION RATIO 30% AND SNR 10 DB. RUNTIME IS MEASURED IN SECONDS.

Image / metric	CP-GAMP	FBCP	TC-AMP
Airplane NMSE	**-21.17**	-19.62	-18.20
Airplane runtime	24.15	64.64	**19.20**
Baboon NMSE	**-15.13**	-15.08	-13.35
Baboon runtime	47.41	122.05	**19.01**
Barbara NMSE	**-16.39**	-15.76	-14.98
Barbara runtime	**24.41**	66.39	29.68
Facade NMSE	**-21.43**	-21.21	-18.47
Facade runtime	18.99	43.32	**15.54**
House NMSE	**-19.95**	-19.37	-16.71
House runtime	22.99	96.26	**15.95**
Sailboat NMSE	**-15.96**	-15.57	-13.70
Sailboat runtime	**19.03**	127.87	20.20

TABLE II
NMSE (DB) VERSUS PRUNING THRESHOLD ON 100^3 TENSORS WITH 20% OBSERVED ENTRIES.

Threshold	R=5	R=10	R=15	R=20	R=25
0.3	-29.82	-27.28	-25.83	-24.52	-23.53
0.1	-29.57	-26.93	-25.45	-24.07	-22.95
0.0	-29.50	-26.91	-25.41	-24.04	-22.91

TABLE III
NMSE (DB) UNDER MODEL MISMATCH ON 100^3 TENSORS WITH 20% OBSERVED ENTRIES.

Setting	R=5	R=10	R=15	R=20	R=25
Gaussian var. 1	-29.82	-27.28	-25.83	-24.52	-23.53
Gaussian var. 2	-28.04	-25.75	-24.45	-23.33	-22.57
Gaussian var. 4	-26.45	-24.59	-23.53	-22.72	-22.25
Laplace var. 1	-27.69	-27.08	-25.62	-22.70	-22.72

according to the observation ratio. We report normalized mean-squared error (NMSE),

$$\mathrm{NMSE} = 20 \log_{10} \frac{\|\hat{\boldsymbol{\mathcal{Z}}} - \boldsymbol{\mathcal{Z}}\|_F}{\|\boldsymbol{\mathcal{Z}}\|_F}. \tag{28}$$

Unless otherwise stated, synthetic experiments use $100 \times 100 \times 100$ tensors, SNR 10 dB, and 10 independent trials. Adaptive CP-GAMP uses maximum iteration 1000, stopping threshold 3×10^{-4}, initial $v^w = 1$, initial $\lambda_r = 0.5$, and pruning threshold $T_\lambda = 0.3$. FBCP [9] and TC-AMP [20] are configured using their published stopping criteria. TC-AMP is given the true rank because it does not learn rank automatically.

B. Synthetic Tensor Reconstruction

Fig. 2 compares adaptive CP-GAMP with FBCP and TC-AMP. Adaptive CP-GAMP and FBCP achieve comparable NMSE across tensor dimensions, ranks, and observation ratios. TC-AMP is less accurate in these tests, which is consistent with its unfolded representation not enforcing the shared CP component structure. The main advantage of CP-GAMP is runtime, since it avoids the matrix inversions required by VI-based updates and scales linearly per iteration in the number of tensor entries.

C. Image Inpainting

We also evaluate image inpainting by treating six 256×256 images in [9] as third-order tensors. On the facade image in Fig. 3, adaptive CP-GAMP reaches NMSE -21.43 dB in 18.99 seconds, compared with -21.21 dB in 43.32 seconds for FBCP and -18.47 dB in 15.54 seconds for TC-AMP. In these six image tests, CP-GAMP obtains the lowest NMSE in the table while remaining faster than FBCP.

Table I shows that CP-GAMP is not always the fastest method on this 256×256 image set because TC-AMP has a lighter unfolded representation. However, TC-AMP pays for this speed with lower reconstruction quality in these tests. FBCP is competitive in NMSE but is slower on all images because its variational updates repeatedly manipulate posterior covariance terms. The CP-GAMP trade-off is therefore favorable in this image setting when both reconstruction fidelity and runtime matter.

D. State Evolution

Fig. 4 compares SE-predicted NMSE curves with empirical CP-GAMP trajectories run without EM hyperparameter learning or pruning. The curves capture the main convergence trend and the effect of the observation ratio.

E. Ablation and Robustness

Tables II and III address robustness to pruning thresholds and model mismatch. The pruning threshold has only a small effect on reconstruction NMSE over the tested range. Prior-variance mismatch and variance-matched Laplace noise degrade performance gracefully, usually within a few dB.

F. Discussion

Across the tested settings, CP-GAMP retains rank and noise learning and factor uncertainty while reducing runtime relative to FBCP. Its NMSE advantage over TC-AMP is consistent with preserving CP component sharing instead of unfolding the tensor. The SE curves describe qualitative behavior only: because the comparison fixes rank and hyperparameters, residual gaps may arise from finite size and Gaussian-approximation error.

The implementation evaluates only AWGN, for which (8)-(9) are closed form. Other separable channels retain the GAMP interface but require task-specific moments. Hence, quadrature or lookup tables add per-entry cost. Their implementation and real-data validation are left for future work.

G. Limitations

CP-GAMP assumes low-rank CP structure. Strong mode-specific subspaces or non-multilinear structure may require Tucker [24], tensor-train [4], or neural models [38]. Its SE-replica analysis is formal, asymptotic, and matched-model, not a finite-sample result for adaptive BG-EM. Experiments

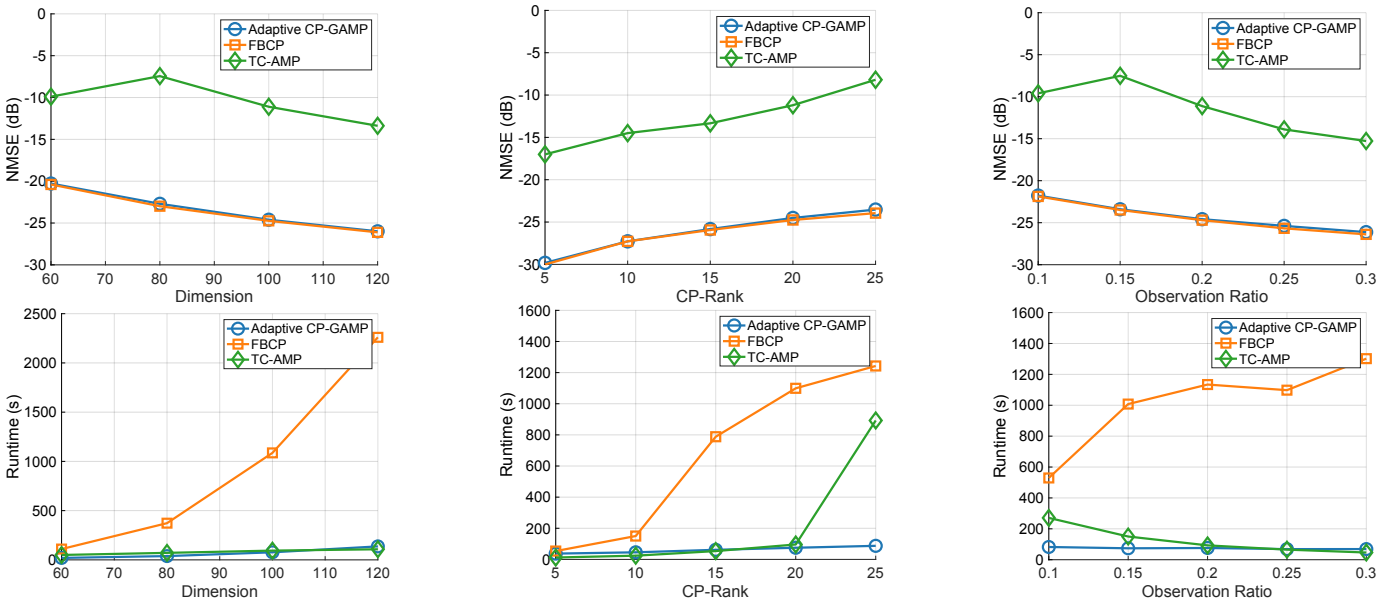


Fig. 2. Synthetic tensor reconstruction at 10 dB SNR. Top: NMSE. Bottom: runtime. Left: varying dimension with rank 20 and observation ratio 0.2. Middle: varying rank for 100^3 tensors and observation ratio 0.2. Right: varying observation ratio for 100^3 tensors with rank 20.

(a) Truth (b) Obs. (c) CP-GAMP (d) FBCP (e) TC-AMP

Fig. 3. Image inpainting with observation ratio 30% and SNR 10 dB.

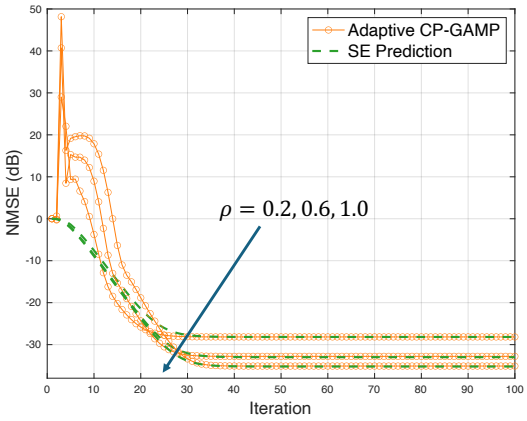


Fig. 4. SE predictions and empirical CP-GAMP trajectories without EM hyperparameter learning or pruning for 100^3 rank-20 tensors at 10 dB SNR under observation ratios $\rho \in \{0.2, 0.6, 1.0\}$.

implement AWGN and the BG denoiser. Other channels and priors require new scalar moments and, for pruning, suitable hierarchical rules.

VII. Conclusion

This paper presented CP-GAMP, a scalable Bayesian CP tensor reconstruction algorithm with EM-based rank and noise learning. The method avoids matrix inversions, keeps the CP multilinear structure, and supports incomplete noisy observations. We derived a formal SE recursion and related its fixed points to replica-symmetric saddle point equations under explicit large-system and matched-model assumptions. Experiments show competitive reconstruction accuracy and lower runtime in the tested settings, and ablations suggest robustness to pruning thresholds and moderate model mismatch.

Appendix A
Derivation for CP-GAMP

This appendix summarizes the derivation that leads from loopy belief propagation to the update equations used in Section IV [33]–[35]. For an observation index $\boldsymbol{i} = (i_1, \ldots, i_N)$, let $\mathcal{V}(\boldsymbol{i}) = \{(\ell, k) : \ell = 1, \ldots, N,\ k = 1, \ldots, R\}$ be the set of adjacent scalar factor variables. The pair $\alpha = (\ell, k) \in \mathcal{V}(\boldsymbol{i})$ denotes $a_{\boldsymbol{i},\alpha} = a^{(\ell)}_{i_\ell,k}$. Let $\Delta^t_{\boldsymbol{i}\to i_n,r}(a^{(n)}_{i_n,r})$ denote the log-message from likelihood node $\boldsymbol{i}$ to variable node $a^{(n)}_{i_n,r}$, and let $\Delta^t_{i_n,r\to\boldsymbol{i}}(a^{(n)}_{i_n,r})$ denote the reverse message. For a neighboring variable index α, write $m^t_{\alpha\to\boldsymbol{i}}(a_{\boldsymbol{i},\alpha}) = \exp\{\Delta^t_{\alpha\to\boldsymbol{i}}(a_{\boldsymbol{i},\alpha})\}$. The sum-product updates in log form are

$$\Delta^t_{\boldsymbol{i}\to i_n,r}(a) = \log \int p(y_{\boldsymbol{i}}|z_{\boldsymbol{i}}) \prod_{\alpha\in\mathcal{V}(\boldsymbol{i})\setminus\{(n,r)\}} m^t_{\alpha\to\boldsymbol{i}}(a_{\boldsymbol{i},\alpha}) \, d\mathbf{a}_{\boldsymbol{i},-(n,r)} + \mathcal{C}, \tag{29}$$

$$\Delta^{t+1}_{i_n,r\to\boldsymbol{i}}(a) = \log p(a) + \sum_{\boldsymbol{i}'\in\mathcal{F}^{(n)}_{i_n,r}\setminus\boldsymbol{i}} \Delta^t_{\boldsymbol{i}'\to i_n,r}(a) + \mathcal{C}. \tag{30}$$

Here $d\mathbf{a}_{\boldsymbol{i},-(n,r)}$ integrates over all $a_{\boldsymbol{i},\alpha}$ with $\alpha \neq (n,r)$, and $\mathcal{C}$ denotes terms independent of the target scalar a. The integral in (29) is high-dimensional because $z_{\boldsymbol{i}} = \sum_r \prod_n a^{(n)}_{i_n,r}$ depends on all neighboring factor variables.

For the target component r and mode n, we use the notation $\hat{a}_{\boldsymbol{i},r}^{(\backslash n)} = \prod_{\ell \neq n} \hat{a}_{i_\ell,r}^{(\ell)}$ for the product of the other-mode means in the same rank-one term. The corresponding product variance is $\omega_{\boldsymbol{i},r}^{(\backslash n)}$ in (14).

Under Assumption 1, the contribution of all variables except $a_{i_n,r}^{(n)}$ is approximated as Gaussian. Conditioned on $a_{i_n,r}^{(n)} = a$, write

$$z_{\boldsymbol{i}} \approx a\, \hat{a}_{\boldsymbol{i},r}^{(\backslash n)} + \hat{p}_{\boldsymbol{i}}^{(\backslash n,r)} + \epsilon_{\boldsymbol{i},i_n,r}, \quad \epsilon_{\boldsymbol{i},i_n,r} \sim \mathcal{N}(0, v_{\boldsymbol{i},i_n,r}^{p}). \tag{31}$$

Here $\hat{p}_{\boldsymbol{i}}^{(\backslash n,r)}$ and $v_{\boldsymbol{i},i_n,r}^{p}$ collect the mean and variance of the aggregate contribution not represented by the argument a of the target factor entry $a_{i_n,r}^{(n)}$ times $\hat{a}_{\boldsymbol{i},r}^{(\backslash n)}$. This reduces the factor-to-variable message to the scalar function

$$H(\hat{p}, v^p; y_{\boldsymbol{i}}) = \log \int p(y_{\boldsymbol{i}}|z)\mathcal{N}(z; \hat{p}, v^p)\, dz. \tag{32}$$

Expanding H to second order around the current estimate gives the AMP residual terms

$$\hat{s}_{\boldsymbol{i}} = \frac{\partial H}{\partial \hat{p}}, \quad -v_{\boldsymbol{i}}^{s} = \frac{\partial^2 H}{\partial \hat{p}^2}, \tag{33}$$

which are equal to (10) for the Gaussian output channel. Let $c_{\boldsymbol{i},r}^{(n)} = \hat{a}_{\boldsymbol{i},r}^{(\backslash n)}$ and expand the scalar factor message as the target entry is perturbed from its current estimate:

$$\hat{p}_{\boldsymbol{i}}(a) \approx \hat{p}_{\boldsymbol{i}} + (a - \hat{a}_{i_n,r}^{(n)}) c_{\boldsymbol{i},r}^{(n)}. \tag{34}$$

The Taylor approximation of the contribution from observation $\boldsymbol{i}$ is therefore, up to constants independent of a,

$$(a - \hat{a}_{i_n,r}^{(n)}) c_{\boldsymbol{i},r}^{(n)} \hat{s}_{\boldsymbol{i}} - \frac{1}{2}(a - \hat{a}_{i_n,r}^{(n)})^2 \left(c_{\boldsymbol{i},r}^{(n)}\right)^2 v_{\boldsymbol{i}}^{s}. \tag{35}$$

Summing these quadratic terms over $\mathcal{F}_{i_n,r}^{(n)}$ gives the edge-dependent Gaussian likelihood

$$\left(\tilde{v}_{i_n,r}^{q}\right)^{-1} = \sum_{\boldsymbol{i} \in \mathcal{F}_{i_n,r}^{(n)}} \left(c_{\boldsymbol{i},r}^{(n)}\right)^2 v_{\boldsymbol{i}}^{s}, \tag{36}$$

$$\tilde{q}_{i_n,r} = \hat{a}_{i_n,r}^{(n)} + \tilde{v}_{i_n,r}^{q} \sum_{\boldsymbol{i} \in \mathcal{F}_{i_n,r}^{(n)}} c_{\boldsymbol{i},r}^{(n)} \hat{s}_{\boldsymbol{i}}. \tag{37}$$

Equations (11)-(12) are the node-aggregated version of these expressions with the first-order Onsager correction retained. Substituting the quadratic approximation into (30) and completing the square yields a Gaussian effective likelihood for each factor variable,

$$\Delta_{i_n,r}^{t+1}(a) \approx \log p(a) - \frac{(a - \hat{q}_{i_n,r})^2}{2 v_{i_n,r}^{q}} + \mathcal{C}. \tag{38}$$

Taking the posterior mean and variance under this scalar density gives the denoising step (15). The Onsager correction in (12) appears from removing the self-interaction between the current variable and the likelihood nodes that sent messages to it. This correction distinguishes CP-GAMP from a naive Gaussian loopy belief-propagation (BP) approximation and provides the AMP-style self-feedback cancellation used by the formal SE derivation.

The expressions used in the main text are obtained after replacing edge-dependent quantities by node-dependent ones. For example, an edge-wise Gaussian approximation would assign a different edge-dependent value $\hat{a}_{i_n,r\to\boldsymbol{i}}^{(n)}$ to every edge between a factor variable and an observation node. Keeping all such edge quantities would require memory proportional to $R|\Omega|N$, which is impractical for large tensors. The AMP expansion suggests that, in the large-system scaling, the difference between an edge quantity and its node aggregate is a lower-order correction once the Onsager term is retained. Thus CP-GAMP stores only $\hat{a}_{i_n,r}^{(n)}$ and $v_{i_n,r}^{a^{(n)}}$ for every factor entry and uses the residual correction to compensate for the removed self-feedback.

For the AWGN output channel, the derivatives of H can be evaluated without numerical integration. Combining the Gaussian pseudo-prior $z_{\boldsymbol{i}} \sim \mathcal{N}(\hat{p}_{\boldsymbol{i}}, v_{\boldsymbol{i}}^{p})$ with $y_{\boldsymbol{i}}|z_{\boldsymbol{i}} \sim \mathcal{N}(z_{\boldsymbol{i}}, v^w)$ gives

$$p(z_{\boldsymbol{i}}|y_{\boldsymbol{i}}, \hat{p}_{\boldsymbol{i}}, v_{\boldsymbol{i}}^{p}) = \mathcal{N}\left(z_{\boldsymbol{i}}; \frac{y_{\boldsymbol{i}} v_{\boldsymbol{i}}^{p} + \hat{p}_{\boldsymbol{i}} v^w}{v_{\boldsymbol{i}}^{p} + v^w}, \frac{v_{\boldsymbol{i}}^{p} v^w}{v_{\boldsymbol{i}}^{p} + v^w}\right). \tag{39}$$

For a missing entry, the likelihood is uninformative and the posterior equals the pseudo-prior. These two cases yield (8)-(9). Other separable channels can be used by replacing this scalar posterior computation while leaving the factor-side denoising unchanged.

Appendix B
EM Updates

The EM updates in (20)-(21) follow from maximizing the expected complete-data log-likelihood [22]. For the activity probability λ_r, the relevant complete-data term is

$$\sum_{n=1}^{N} \sum_{i_n=1}^{I_n} \log\left((1-\lambda_r)\delta(a_{i_n,r}^{(n)}) + \lambda_r \mathcal{N}(a_{i_n,r}^{(n)}; 0, 1)\right). \tag{40}$$

Let $b_{i_n,r}^{(n)}$ be the latent Bernoulli indicator that equals one when $a_{i_n,r}^{(n)}$ is active. The scalar posterior in CP-GAMP gives

$$\mathbb{E}[b_{i_n,r}^{(n)}|\boldsymbol{\mathcal{Y}}] = \pi(\hat{q}_{i_n,r}, v_{i_n,r}^{q}; \lambda_r^j). \tag{41}$$

The expected log-likelihood for λ_r is therefore

$$Q(\lambda_r) = \sum_{n,i_n} \Big[\pi(\hat{q}_{i_n,r}, v_{i_n,r}^{q}; \lambda_r^j) \log \lambda_r + \left(1 - \pi(\hat{q}_{i_n,r}, v_{i_n,r}^{q}; \lambda_r^j)\right) \log(1-\lambda_r)\Big] + \mathcal{C}. \tag{42}$$

Setting $\partial Q / \partial \lambda_r = 0$ yields (20).

For the noise variance, the complete-data term is

$$-\frac{1}{2} \sum_{\boldsymbol{i} \in \Omega} \left[\log v^w + \frac{|y_{\boldsymbol{i}} - z_{\boldsymbol{i}}|^2}{v^w}\right]. \tag{43}$$

The posterior expectation of the squared residual is

$$\mathbb{E}\left[|y_{\boldsymbol{i}} - z_{\boldsymbol{i}}|^2 |\boldsymbol{\mathcal{Y}}\right] = |y_{\boldsymbol{i}} - \hat{z}_{\boldsymbol{i}}|^2 + v_{\boldsymbol{i}}^{z}. \tag{44}$$

Maximizing with respect to v^w gives the average residual energy over observed entries, which is exactly (21).

APPENDIX C
RANK LEARNING AND PRUNING

The BG prior introduces a component-level activity probability λ_r shared by the r-th columns of all factor matrices. This differs from placing independent sparsity parameters on every factor entry. The shared parameter encourages entire rank-one components to be removed, which is the desired behavior for CP-rank learning. At an EM iteration, λ_r is updated by averaging scalar posterior activity probabilities over all entries in component r:

$$\lambda_r^{j+1} = \frac{1}{\sum_n I_n} \sum_n \sum_{i_n} \pi(\hat{q}_{i_n,r}, v^q_{i_n,r}; \lambda_r^j). \tag{45}$$

Thus a component remains active only if it is consistently supported across modes and factor rows. This averaging makes the rank estimate less sensitive to isolated noisy entries than a rule based on individual coefficient magnitudes.

The pruning rule $\lambda_r < T_\lambda$ is not required to define the posterior approximation. It is used to reduce computation after EM has already assigned low posterior activity to a component. Without pruning, inactive components remain in the factor matrices with means close to zero but still contribute to the cost of computing (6)-(7). Pruning therefore acts as a computational shortcut and as a way to report an explicit effective rank. The ablation in Table II shows that the reconstruction NMSE changes little when the threshold is reduced from 0.3 to 0.1 or removed entirely, suggesting that the posterior shrinkage is doing most of the statistical work and pruning mainly affects model compactness.

There is also a complementary view of rank learning based on rank identifiability and moment matching in [39]. Such approaches can provide an initial estimate or upper bound for the rank before message passing starts. In CP-GAMP, this information could be used to choose the maximum rank R or to initialize λ_r non-uniformly. The BG-EM mechanism would then refine the rank estimate during inference. This combination may reduce the amount of over-parameterization needed in large tensors and is a useful direction for future work.

APPENDIX D
SE-REPLICA CORRESPONDENCE

This appendix gives the derivation behind Proposition 1. The argument follows the empirical-convergence and replica-symmetric strategy used for ML-BiGAMP [36], with the bilinear product replaced by the CP multilinear product. For $N = 2$ this collapses to the ML-BiGAMP-type bilinear case, but the extension to arbitrary order is nontrivial because the mode-wise overlaps and Onsager corrections must be tracked separately.

SE derivation. Assume empirical convergence with bounded second moments in the pseudo-Lipschitz sense used in AMP SE [17]. For fixed mode n, the CP-GAMP input to $a^{(n)}_{i_n,r}$ is a sum over $O(\prod_{\ell\neq n} I_\ell)$ neighboring entries. After the Onsager correction in (12), the large-system central limit theorem (CLT) motivates the scalar channel

$$\hat{q}^{(n)} = a^{(n)} + \sqrt{\Sigma_a^{(n)}}\xi, \quad \xi \sim \mathcal{N}(0,1), \tag{46}$$

$$q_a^{(n)} = \mathbb{E}\left[\left(\mathbb{E}[a^{(n)}|\hat{q}^{(n)}]\right)^2\right]. \tag{47}$$

Under the matched prior, the Nishimori identity [40] makes this second moment equal to the planted-estimate overlap. The CP product then gives

$$V = \mathbb{E}[(z_{\boldsymbol{i}} - \hat{z}_{\boldsymbol{i}})^2] = \chi_z - R\prod_{n=1}^N q_a^{(n)}. \tag{48}$$

To see this product form, write the contribution of component r as $x_{\boldsymbol{i},r} = \prod_n a^{(n)}_{i_n,r}$ and its estimate as $\hat{x}_{\boldsymbol{i},r} = \prod_n \hat{a}^{(n)}_{i_n,r}$. The zero-mean independent components make the cross terms $\mathbb{E}[x_{\boldsymbol{i},r}x_{\boldsymbol{i},r'}]$ and $\mathbb{E}[x_{\boldsymbol{i},r}\hat{x}_{\boldsymbol{i},r'}]$ vanish for $r \neq r'$. For a fixed component, empirical convergence gives

$$\mathbb{E}[x_{\boldsymbol{i},r}\hat{x}_{\boldsymbol{i},r}] = \prod_{n=1}^N \mathbb{E}[a^{(n)}\hat{a}^{(n)}] = \prod_{n=1}^N q_a^{(n)}, \tag{49}$$

where the last equality uses the matched-prior Nishimori relation. Thus the tensor-level overlap is the product of the mode-wise overlaps, multiplied by the number of active rank components. For the AWGN missing-data channel, $v^p_{\boldsymbol{i}} \to V$ and averaging over the Bernoulli mask yields

$$v^z = \rho\frac{v^w V}{v^w + V} + (1-\rho)V, \tag{50}$$

$$v^s = \frac{1}{V}\left(1 - \frac{v^z}{V}\right) = \frac{\rho}{v^w + V}, \tag{51}$$

$$\frac{1}{\Sigma_a^{(n)}} = \frac{\rho\prod_{\ell\neq n} I_\ell q_a^{(\ell)}}{v^w + V}. \tag{52}$$

These identities are equivalent to (23)-(26).

Replica correspondence. The corresponding formal replica calculation [37] starts from the quenched log partition function, normalized according to the chosen large-system scaling, and applies the identity

$$\mathbb{E}\log p(\boldsymbol{\mathcal{Y}}) = \lim_{\tau\to 0} \partial_\tau \log \mathbb{E}p^\tau(\boldsymbol{\mathcal{Y}}). \tag{53}$$

For integer τ, the replicated factors share the same observation tensor. Introduce overlaps

$$Q_{ab}^{(n)} = \frac{1}{I_n R}\sum_{i_n,r} a^{(n)(a)}_{i_n,r} a^{(n)(b)}_{i_n,r}, \quad a,b = 0,\ldots,\tau, \tag{54}$$

where $a = 0$ denotes the planted replica. The CP product gives replicated tensor covariance $R\prod_n Q_{ab}^{(n)}$. Under the replica-symmetric (RS) ansatz, $Q_{aa}^{(n)} = \chi_a^{(n)}$ and $Q_{ab}^{(n)} = q_a^{(n)}$ for $a \neq b$; the conjugate saddle-point variables decouple each factor entry into the same scalar AWGN channel as SE. Stationarity gives

$$q_a^{(n)} = \mathbb{E}\left[\left(\mathbb{E}[a^{(n)}|a^{(n)} + \sqrt{\Sigma_a^{(n)}}\xi]\right)^2\right], \tag{55}$$

$$\Sigma_a^{(n)} = \frac{v^w + V}{\rho \prod_{\ell \neq n} I_\ell q_a^{(\ell)}}, \tag{56}$$

$$V = \chi_z - R \prod_{n=1}^{N} q_a^{(n)}. \tag{57}$$

These match the fixed-point equations of (23)-(26). This is the fixed-point correspondence summarized in Proposition 1. The same V is the SE per-entry tensor error, $V = \mathbb{E}[(z_{\boldsymbol{i}} - \hat{z}_{\boldsymbol{i}})^2]$. At the replica-symmetric Bayes branch it gives the formal MMSE prediction, with normalized error V/χ_z. Thus the SE curve is an algorithmic trajectory toward one saddle-point branch, while the lowest-error Bayes branch gives the formal MMSE benchmark. The argument relies on replica symmetry, matched priors, and convergence to the informative branch.

The finite-size gap in the no-EM SE comparison may arise because the factor graph has many short cycles and the Gaussian approximation relies on averaging over many weakly dependent contributions.

Appendix E
Experimental Settings and Reproducibility

In our experiments, we use random initialization unless otherwise stated. The pruning threshold $T_\lambda = 0.3$ is applied only after EM has updated λ_r, which reduces unnecessary components and accelerates later iterations.

For synthetic experiments, a ground-truth tensor is generated by drawing all factor entries independently from $\mathcal{N}(0, 1)$ and then forming the CP tensor in (1). The noise variance is selected to match the target SNR,

$$\mathrm{SNR} = 10 \log_{10} \frac{\mathrm{Var}(\boldsymbol{\mathcal{Z}})}{v^w}. \tag{58}$$

The observation mask is sampled independently with probability ρ for each entry. All reported synthetic NMSE values are averages over 10 independent tensors and masks.

The compared methods are configured as follows. For synthetic experiments, both adaptive CP-GAMP and FBCP use the initial working rank $R_{\mathrm{init}} = \max(\lfloor \max_n I_n/2 \rfloor, R_{\mathrm{true}})$, where R_{true} is the ground-truth CP rank. CP-GAMP then learns the active components through λ_r. FBCP is run with automatic rank determination, following the published stopping rule. TC-AMP is given the true rank because it does not have the same EM rank-learning mechanism. This gives TC-AMP extra rank information and makes the comparison conservative with respect to rank selection. The maximum iteration count is 1000 for all iterative methods. The stopping threshold for CP-GAMP is 3×10^{-4} on the relative change of the reconstructed tensor.

For image inpainting, each red-green-blue (RGB) image is represented as a $256 \times 256 \times 3$ tensor. The missing mask is applied independently to tensor entries, and AWGN is added at 10 dB SNR before masking. Runtime is measured wall-clock time for the full reconstruction, including hyperparameter updates. Synthetic runs used an Intel i5-8500 desktop and image runs used an Apple M2 Pro laptop, both with 16 GB memory. Because hardware differs, runtime comparisons should be interpreted within each experimental group rather than across synthetic and image tasks.

Appendix F
Observation Channels and Priors

CP-GAMP separates multilinear aggregation from scalar inference. The output module requires the posterior mean and variance of $z_{\boldsymbol{i}}$ under a Gaussian pseudo-prior and the derivatives summarized by $\hat{s}_{\boldsymbol{i}}$ and $v^s_{\boldsymbol{i}}$. These are closed form for AWGN. Unimplemented alternatives include Bernoulli, count, and heavy-tailed likelihoods [16]. Their moments require closed forms, lookup tables, or quadrature, with K-point quadrature adding $O(K|\Omega|)$ work per iteration.

For any separable prior $p_0(a)$, the factor module uses the scalar channel $\hat{q} = a + w$, $w \sim \mathcal{N}(0, v^q)$, and returns its posterior mean and variance. Gaussian, BG, and some mixture priors are closed form; nonconjugate priors require numerical evaluation. The shared BG activity λ_r supplies rank shrinkage, whereas other priors need an analogous hierarchy for pruning, and their SE characterization must be rederived.

Appendix G
Complexity Comparison

This appendix gives a more explicit comparison between CP-GAMP and the main baselines. Let N be the tensor order, R the working rank, and Ω the observed entry set. For CP-GAMP, the dominant cost per iteration comes from computing the multilinear predictions and residual statistics over Ω, followed by the scalar denoising updates over all factor entries. This gives the practical cost

$$O\left(NR|\Omega| + R \sum_{n=1}^{N} I_n \right), \tag{59}$$

where the first term dominates in large incomplete tensors. The memory footprint is similarly modest because the algorithm stores scalar means and variances for factor entries and scalar residual terms for observations.

For FBCP [9], the cost structure is different. In addition to expectation updates over observed entries, VI requires covariance updates for latent factors and repeated products involving Khatri-Rao structure. Using the same notation, its per-iteration cost is

$$O\left(NR^2|\Omega| + R^3 \sum_{n=1}^{N} I_n \right). \tag{60}$$

These terms reflect rank-dependent sufficient-statistic accumulation, dense covariance updates, and matrix inversions. Even when efficient implementations reduce constant factors, the resulting runtime grows much faster with R than the scalar-denoiser structure of CP-GAMP. This explains the widening runtime gap in Fig. 2 as the rank or tensor dimension increases.

TC-AMP [20] is closer to CP-GAMP in spirit because it also avoids VI-style covariance updates. Its main limitation is not computational but structural. The unfolding step transforms

the tensor problem into a matrix factorization problem and therefore weakens the shared-component interpretation of CP factors across all modes. This is why TC-AMP can remain fast on the 256×256 image inpainting tasks while showing lower NMSE accuracy than direct CP-GAMP updates in our tests.

REFERENCES

[1] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky, "Tensor decompositions for learning latent variable models," *J. Mach. Learn. Res.*, vol. 15, pp. 2773–2832, 2014.

[2] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3551–3582, 2017.

[3] A. H. Williams, T. H. Kim, F. Wang, S. Vyas, S. I. Ryu, K. V. Shenoy, M. Schnitzer, T. G. Kolda, and S. Ganguli, "Unsupervised discovery of demixed, low-dimensional neural dynamics across multiple timescales through tensor component analysis," *Neuron*, vol. 98, no. 6, pp. 1099–1115.e8, 2018.

[4] L. Xu, L. Cheng, N. Wong, and Y.-C. Wu, "Overfitting avoidance in tensor train factorization and completion: Prior analysis and inference," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, 2021, pp. 1439–1444.

[5] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis," *UCLA Work. Papers in Phonetics*, vol. 16, pp. 1–84, 1970.

[6] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[7] W. Chu and Z. Ghahramani, "Probabilistic models for incomplete multi-dimensional arrays," in *Proc. Int. Conf. Artif. Intell. Statist.*, ser. Proc. Mach. Learn. Res., vol. 5. PMLR, 2009, pp. 89–96.

[8] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell, "Temporal collaborative filtering with Bayesian probabilistic tensor factorization," in *Proc. SIAM Int. Conf. Data Mining*, 2010, pp. 211–222.

[9] Q. Zhao, L. Zhang, and A. Cichocki, "Bayesian CP factorization of incomplete tensors with automatic rank determination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1751–1763, 2015.

[10] P. Rai, Y. Wang, S. Guo, G. Chen, D. Dunson, and L. Carin, "Scalable Bayesian low-rank decomposition of incomplete multiway tensors," in *Proc. Int. Conf. Mach. Learn.*, ser. Proc. Mach. Learn. Res., vol. 32, no. 2. PMLR, 2014, pp. 1800–1808.

[11] S. Budzinskiy and N. Zamarashkin, "Variational Bayesian inference for CP tensor completion with subspace information," *Lobachevskii J. Math.*, vol. 44, no. 8, pp. 3016–3027, 2023.

[12] S. Fang, A. Narayan, R. Kirby, and S. Zhe, "Bayesian continuous-time Tucker decomposition," in *Proc. Int. Conf. Mach. Learn.*, ser. Proc. Mach. Learn. Res., vol. 162. PMLR, 2022, pp. 6235–6245.

[13] Z. Tao, T. Tanaka, and Q. Zhao, "Scalable Bayesian tensor ring factorization for multiway data analysis," in *Proc. Int. Conf. Neural Inf. Process.*, ser. Lecture Notes in Computer Science, vol. 14447. Singapore: Springer Nature Singapore, 2024, pp. 490–503.

[14] ——, "Efficient nonparametric tensor decomposition for binary and count data," in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 14, 2024, pp. 15 319–15 327.

[15] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proc. Natl. Acad. Sci.*, vol. 106, no. 45, pp. 18 914–18 919, 2009.

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

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

[18] J. T. Parker, P. Schniter, and V. Cevher, "Bilinear generalized approximate message passing—part I: Derivation," *IEEE Trans. Signal Process.*, vol. 62, no. 22, pp. 5839–5853, 2014.

[19] ——, "Bilinear generalized approximate message passing—part II: Applications," *IEEE Trans. Signal Process.*, vol. 62, no. 22, pp. 5854–5867, 2014.

[20] Y. Li, C. Yin, and Z. Han, "An approximate message passing approach for tensor-based seismic data interpolation with randomly missing traces," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2016, pp. 1402–1406.

[21] J. Kadmon and S. Ganguli, "Statistical mechanics of low-rank tensor decomposition," in *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 8201–8212.

[22] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc. Ser. B Methodol.*, vol. 39, no. 1, pp. 1–22, 1977.

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

[24] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[25] Z. Yang, L. T. Yang, H. Wang, H. Zhao, and D. Liu, "Bayesian non-negative tensor completion with automatic rank determination," *IEEE Trans. Image Process.*, vol. 34, pp. 2036–2051, 2025.

[26] A. Montanari and E. Richard, "A statistical model for tensor PCA," in *Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 2897–2905.

[27] T. Lesieur, L. Miolane, M. Lelarge, F. Krzakala, and L. Zdeborová, "Statistical and computational phase transitions in spiked tensor estimation," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2017, pp. 511–515.

[28] J. Barbier, N. Macris, and L. Miolane, "The layered structure of tensor estimation and its mutual information," in *Proc. 55th Annu. Allerton Conf. Commun., Control, Comput.*, 2017, pp. 1056–1063.

[29] C. Zhang, X. Cao, W. Liu, I. Tsang, and J. Kwok, "Nonparametric iterative machine teaching," in *Proc. 40th Int. Conf. Mach. Learn.*, ser. Proc. Mach. Learn. Res., vol. 202. PMLR, 2023, pp. 40 851–40 870.

[30] C. Zhang, W. Zuo, B. Cheng, Y. Wang, W.-B. Kou, Y.-C. Wu, and N. Wong, "NTK-guided implicit neural teaching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2026.

[31] C. Zhang, W. Bu, Z. Ren, Z. Liu, Y.-C. Wu, and N. Wong, "Nonparametric teaching for graph property learners," in *Proc. 42nd Int. Conf. Mach. Learn.*, ser. Proc. Mach. Learn. Res., vol. 267. PMLR, 2025, pp. 74 515–74 539.

[32] C. Zhang, J. Wang, B. Cheng, Z. Chen, W. Xu, C. Wang, M. Canini, F. Orabona, Y.-C. Wu, and N. Wong, "Nonparametric teaching of attention learners," in *Proc. Int. Conf. Learn. Represent.*, 2026.

[33] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[34] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[35] B. J. Frey and D. J. C. MacKay, "A revolution: Belief propagation in graphs with cycles," in *Adv. Neural Inf. Process. Syst.*, vol. 10, 1997, pp. 479–485.

[36] Q. Zou, H. Zhang, and H. Yang, "Multi-layer bilinear generalized approximate message passing," *IEEE Trans. Signal Process.*, vol. 69, pp. 4529–4543, 2021.

[37] M. Mézard, G. Parisi, and M. A. Virasoro, *Spin Glass Theory and Beyond: An Introduction to the Replica Method and Its Applications*, ser. World Scientific Lecture Notes in Physics. Singapore: World Scientific, 1987, vol. 9.

[38] H. Liu, Y. Li, M. Tsang, and Y. Liu, "CoSTCo: A neural tensor completion model for sparse tensors," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 324–334.

[39] E. da Silva, A. Klami, D. Mesquita, and I. Urteaga, "On the identifiability of tensor ranks via prior predictive matching," 2025, arXiv:2510.14523.

[40] H. Nishimori, *Statistical Physics of Spin Glasses and Information Processing: An Introduction*, ser. International Series of Monographs on Physics. Oxford, U.K.: Oxford University Press, 2001, vol. 111.