Sparse Regression LDPC Codes

Jamison R. Ebert, *Member, IEEE*, Jean-Francois Chamberland, *Senior Member, IEEE*, and Krishna R. Narayanan, *Fellow, IEEE*

***Abstract*— This article introduces a novel concatenated coding scheme called sparse regression LDPC (SR-LDPC) codes. An SR-LDPC code consists of an outer non-binary LDPC code and an inner sparse regression code (SPARC), whose respective field size and section sizes are equal. For such codes, an efficient decoding algorithm is proposed based on approximate message passing (AMP) that dynamically shares soft information between inner and outer decoders. This dynamic exchange of information is facilitated by a denoiser that runs belief propagation (BP) on the factor graph of the outer LDPC code within each AMP iteration. It is shown that this BP denoiser falls within the framework of non-separable denoising functions and subsequently, that state evolution holds for the proposed AMP-BP algorithm. Leveraging the rich structure of SR-LDPC codes, this article proposes an efficient low-dimensional approximate state evolution recursion that can be used for efficient hyperparameter tuning, thus paving the way for future work on optimal code design. Finally, numerical simulations demonstrate that SR-LDPC codes outperform contemporary codes over the AWGN channel for parameters of practical interest. SR-LDPC codes are shown to be viable means for obtaining shaping gains over the AWGN channel.**

***Index Terms*— LDPC codes, sparse regression codes (SPARCs), approximate message passing, belief propagation, shaping gain.**

I. Introduction

LOW-DENSITY parity check (LDPC) codes have been studied extensively over the past several decades [1], [2], [3], [4], [5], [6], [7], [8] and are known to be capacity approaching over the additive white Gaussian noise (AWGN) channel. Furthermore, under certain conditions, encoded messages can be recovered efficiently using iterative belief propagation (BP) decoding. Since the complexity per iteration of BP decoding grows linearly with the block length, this paradigm offers a pragmatic solution for decoding codes with long block lengths [9]. Moreover, some spatially coupled LDPC constructions feature capacity approaching iterative decoding thresholds while also avoiding the pitfall of error floors [10], [11], [12], [13], [14], [15]. However, systems operating at shorter block lengths may not be conducive to the application of spatial coupling. In such situations, non-binary LDPC codes have been leveraged as means to provide adequate performance [6], [7], [16], [17], [18].

In a seemingly unrelated research direction, Joseph and Barron introduce the concept of a sparse regression code (SPARC) [19], [20], [21] which establishes a connection between code design and sparse recovery in high dimensions. SPARC codewords consist of sparse linear combinations of the columns of a design matrix; thus, the problem of SPARC decoding is equivalent to that of noisy support recovery for which many low complexity frameworks have been studied in the literature. Most notably, Barbier et al. introduce an approximate message passing (AMP) decoder for SPARCs in [22]. It is shown that SPARCS with AMP decoding achieves the asymptotic single-user AWGN channel capacity under an appropriately chosen power allocation [23], [24] or when combined with spatial coupling [25]. There have also been efforts to maximize the finite block-length performance of SPARCs [26], [27].

A popular strategy for improving the performance of codes in practical settings is to adopt a concatenated structure. For example, Greig and Venkataramanan combine a binary LDPC code with a SPARC and propose the following decoding algorithm [26]. First, AMP is run to decode the SPARC; then, the factor graph of the LDPC code is initialized using the soft outputs of AMP; BP is subsequently run to decode the LDPC code; and finally, AMP is run once more to decode the SPARC after removing the contribution of confidently decoded sections. This approach is shown to provide significant performance benefits over uncoded SPARCs for finite block lengths. Similarly in [27], Cao and Vontobel concatenate a SPARC and a cyclic redundancy check (CRC) code for the complex AWGN channel. In this scheme, the outer CRC code serves as an error detection mechanism to determine the true codeword among the multiple candidate paths retained from the static outputs of AMP. Interestingly, the schemes discussed in both [26] and [27] produce a steep waterfall in error performance, a phenomenon that is not achieved by the standalone AMP decoder when operating over short block-lengths. From a theoretic perspective, Liang et al. show that, with a carefully designed outer code, a compressed-coding scheme can asymptotically achieve the single-user Gaussian capacity provided that the state evolution for AMP remains accurate in the presence of the outer code [28]. These results suggest that concatenated schemes involving SPARCs are promising codes for the AWGN channel.

Received 13 November 2023; revised 19 October 2024; accepted 27 October 2024. Date of publication 12 November 2024; date of current version 26 December 2024. This work was supported in part by the National Science Foundation (NSF) under Grant CCF-2131106 and Grant CNS-2148354 and in part by Qualcomm Technologies Inc., through their University Relations Program.An earlier version of this paper was presented in part at the 2023 Information Theory Applications (ITA) Workshop and in part at the 2023 IEEE International Symposium on Information Theory (ISIT) [DOI: 10.1109/ISIT54713.2023.10206818]. *(Corresponding author: Jean-Francois Chamberland.)*

Jamison R. Ebert is with Qualcomm Technologies Inc., San Diego, CA 92121 USA (e-mail: jrebert@tamu.edu).

Jean-Francois Chamberland and Krishna R. Narayanan are with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: chmbrlnd@tamu.edu; krn@tamu.edu).

Communicated by K. R. Duffy, Associate Editor for Coding and Decoding.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIT.2024.3496486.

Digital Object Identifier 10.1109/TIT.2024.3496486

Concatenated structures with SPARC-like inner codes have also been proposed in the context of unsourced random access [29], [30], [31]. In [30], Amalladinne et al. demonstrate that, under AMP decoding, the structure of a judiciously designed outer code can be integrated into the iterative recovery algorithm for the inner code via a dynamic denoising function. Surprisingly, despite being mentioned by Liu et al. in [32] as a possible future research direction, such an approach has not been considered for the single-user scenario. Therefore, the purpose of this article is to address this deficiency.

A. Main Contributions

In this article, a novel concatenated coding scheme is introduced consisting of an outer non-binary LDPC code and an inner sparse regression code, where the field size of the outer code equals the section size of the inner code. An efficient decoding algorithm based on AMP is presented that allows for information to be dynamically shared between inner and outer decoders. This dynamic exchange of information is facilitated by a denoiser that performs BP on the factor graph of the outer LDPC code during each AMP iteration. It is shown that this denoiser falls within the framework of AMP with non-separable denoisers and subsequently, that the state evolution formalism holds. Leveraging the rich mathematical structure inherent in both the code structure and the decoding algorithm, an approximate state evolution recursion is proposed for efficient hyperparameter tuning and code optimization. Finally, the proposed code, referred to as a *Sparse Regression LDPC (SR-LDPC)* [33] code, is shown to outperform other SPARC and LDPC code constructions over the AWGN channel for parameters of practical interest. Numerical results suggest that SR-LDPC codes may be leveraged as means to obtain shaping gain over the AWGN channel.

B. Organization

The remainder of this article is organized as follows. Section II describes the channel model and introduces SR-LDPC encoding and decoding. Section III describes the design of the dynamic denoiser and investigates some of its properties. Section IV utilizes the structure of SR-LDPC codes and the proposed decoding algorithm to develop a low-dimensional approximate state evolution recursion for code optimization. Then, Section V presents numerical simulation results highlighting the benefits of SR-LDPC codes. Finally, Section VI offers concluding remarks. Derivations and proofs for the theorems contained throughout this article may be found in Appendices A and B.

II. System Model

We consider a memoryless point-to-point AWGN channel where both the transmitter and the receiver are equipped with a single antenna. In this model, the received signal $\mathbf{y} \in \mathbb{R}^n$ is given by

$$\mathbf{y} = \mathbf{x} + \mathbf{z}, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the transmitted signal, $\mathbf{z} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \mathbf{I}\right)$ represents AWGN, and n denotes the number of channel uses or, equivalently, the number of (real) degrees of freedom. The signal-to-noise ratio (SNR) is expressed as

$$\frac{E_b}{N_0} = \frac{\mathbb{E}\left[\|\mathbf{x}\|^2\right]}{2B\sigma^2}, \tag{2}$$

where B denotes the number of information bits conveyed in $\mathbf{x}$. The set of codewords is subject to an average power constraint which, without loss of generality, can be set to one (i.e., $\mathbb{E}\left[\|\mathbf{x}\|^2\right] = 1$), with the understanding that a given SNR may be achieved by adjusting the noise variance. As mentioned above, we wish to study a coding architecture comprised of a sparse regression inner code [19], [20], [21], and a non-binary LDPC outer code [6], [7], [8]. We elaborate on the encoding process and the decoding scheme below.

A. SR-LDPC Encoding

The proposed encoding process features a sequence of three distinct steps: q-ary LDPC encoding, indexing of LDPC symbols, and inner CS encoding. In the first step, the information bits are encoded into a q-ary LDPC codeword via well-established operations [6], [7], [8]. The second step transforms the q-ary LDPC codeword into a suitable sparse vector. The last step is the matrix multiplication emblematic of a sparse regression code [19], [20], [21]. We summarize the notions pertaining to this encoding process below while concurrently introducing necessary notation.

1) q-Ary LDPC Encoding: The LDPC encoder takes a binary sequence $\mathbf{w} \in \mathbb{F}_2^B$ as its input and maps it to a q-ary codeword $\mathbf{v} \in \mathbb{F}_q^L$, where q denotes the size of the Galois field [6], [7]. Note that for $\mathbb{F}_q$ to be a field, q must be a power of a prime. Throughout this article, we assume q is of the form $q = 2^m$ for some $m \in \mathbb{N}, m > 1$. We represent the resultant codeword in concatenated form as

$$\mathbf{v} = (v_1, v_2, \ldots, v_L), \tag{3}$$

where the ℓth element v_ℓ lies in finite field $\mathbb{F}_q$ and L is the length of the resulting codeword.

Remark 1: There exists a bijection $\Phi : \mathbb{F}_q \to [q]$ between the elements of $\mathbb{F}_q$ and the integers $[q] = \{0, 1, \ldots, q-1\}$, where the integer 0 represents the zero element of $\mathbb{F}_q$ and the integer 1 represents the unity element of $\mathbb{F}_q$ [7]. Throughout this article, we adopt such an arbitrary, but fixed bijection. We exploit this relation by employing the same variable for a field element $g \in \mathbb{F}_q$ and for its corresponding integer $\Phi(g) \in [q]$. This slight abuse of notation greatly simplifies the exposition of SR-LDPC codes. Furthermore, its use should not lead to confusion because one can unambiguously infer from context whether g refers to the field element or to its integer representation.

2) LDPC Symbol Indexing: With Remark 1 in mind, it becomes straightforward to explain the second step of the encoding process. Coded symbol sparsification/indexing consists of mapping $v_\ell \in \mathbb{F}_q$ to standard basis vector $\mathbf{e}_{v_\ell} \in \mathbb{R}^q$ and subsequently stacking the L basis vectors together. We emphasize that entry ℓ of $\mathbf{v}$ is an element of $\mathbb{F}_q$,

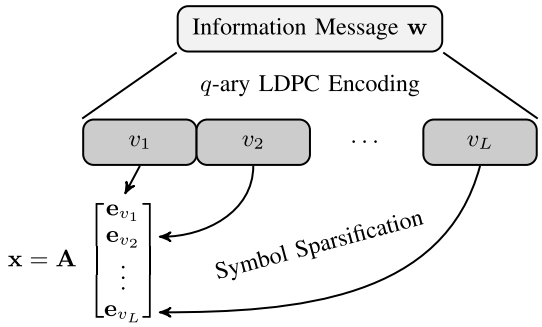

Fig. 1. Depiction of the SR-LDPC encoding process.

but v_ℓ in $\mathbf{e}_{v_\ell}$ refers to an integer in $[q]$ under our overloaded notation. With that, the output of the indexing process becomes

$$\mathbf{s} = \begin{bmatrix} \mathbf{e}_{v_1} \\ \vdots \\ \mathbf{e}_{v_L} \end{bmatrix}, \tag{4}$$

where $\mathbf{s}$ is an L-sparse vector of length qL. Vector $\mathbf{s}$ has a structure akin to that of a sparse regression code prior to multiplication by a large random matrix. This structured sparsity can be exploited during decoding.

3) Inner CS Encoding: The last phase of the encoding process consists in pre-multiplying vector $\mathbf{s}$ by matrix $\mathbf{A}$ to obtain $\mathbf{x} = \mathbf{A}\mathbf{s}$, where $\mathbf{A} \in \mathbb{R}^{n \times qL}$, $n \ll qL$, and $\mathbf{A}_{i,j} \sim \mathcal{N}\left(0, \frac{1}{n}\right)$. Equation (1) may thus be rewritten as:

$$\mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{z}. \tag{5}$$

The overall encoding process is depicted in Fig. 1. We are now ready to discuss the decoding process for SR-LDPC codes.

B. SR-LDPC Decoding

Paralleling the development of AMP for sparse regression codes [24] and drawing inspiration from concatenated AMP systems [28], [32], we wish to create an iterative process to recover state vector $\mathbf{s}$ from $\mathbf{y}$ using AMP. However, a notable distinction between our system model and previously published articles is the presence of a q-ary LDPC outer code. Thus, we wish to create an AMP decoder that simultaneously takes advantage of the structured sparsity in $\mathbf{s}$ and the parity structure embedded in the LDPC outer code. This can be accomplished by incorporating message passing on the factor graph of the LDPC code into the AMP denoiser. A similar approach proposed by Amalladinne et al. can be found in [30], where the intended application is unsourced random access. While the two strategies are conceptually similar, the denoiser we wish to utilize below differs from the one employed in [30] because, in the problem at hand, only one codeword is present within $\mathbf{y}$. This distinction simplifies the structure of the code and enables us to leverage a denoiser that more closely parallels traditional message passing algorithms for q-ary LDPC codes.

Our AMP composite algorithm is as follows,

$$\mathbf{z}^{(t)} = \mathbf{y} - \mathbf{A}\mathbf{s}^{(t)} + \frac{\mathbf{z}^{(t-1)}}{n} \operatorname{div} \boldsymbol{\eta}_{t-1}\left(\mathbf{r}^{(t-1)}\right) \tag{6}$$

$$\mathbf{r}^{(t)} = \mathbf{A}^{\mathrm{T}}\mathbf{z}^{(t)} + \mathbf{s}^{(t)} \tag{7}$$

$$\mathbf{s}^{(t+1)} = \boldsymbol{\eta}_t\left(\mathbf{r}^{(t)}\right), \tag{8}$$

where the superscript t denotes the iteration count. The algorithm is initialized with conditions $\mathbf{s}^{(0)} = \mathbf{0}$ and $\mathbf{z}^{(0)} = \mathbf{y}$. Furthermore, every quantity with a negative iteration count is equal to the zero vector.

Equation (6) computes the *residual* error under the current state estimate $\mathbf{s}^{(t)}$ enhanced with an Onsager correction term. This residual error is used to compute an *effective observation* in (7), which is passed through a *denoiser* to produce a revised *state estimate* in (8). The denoising functions $(\boldsymbol{\eta}_t(\cdot))_{t\geq 0}$ seek to exploit the structure of $\mathbf{s}$ to promote AMP's convergence to the true state.

The performance of AMP as a function of iteration count may be characterized through the state evolution formalism [34], [35], [36]. State evolution is an asymptotic tool, rooted in the analysis of large systems, that captures performance through a Gaussian approximation. Under suitable regularity conditions, this asymptotic approach is valid in that random vectors in the approximate Gaussian model converge in distribution to their counterparts in the original system [37]. The foundation for state evolution in the current setting is the AMP framework for non-separable denoisers put forth by Berthier, Montanari, and Nguyen in [35]. Below, we reproduce a crucial result from [35] which will motivate much of the remainder of this article.

Theorem 2 (Adapted From [35]): Assume that

1) $\mathbf{A}_{i,j} \sim \mathcal{N}\left(0, \frac{1}{n}\right)$ $\forall i \in [n], j \in [qL]$
2) $\forall t, \boldsymbol{\eta}_t : \mathbb{R}^{qL} \to \mathbb{R}^{qL}$ is uniformly Lipschitz in qL
3) The quantity $\frac{\|\mathbf{s}\|_2}{\sqrt{qL}} \to c$ as $qL \to \infty$ for some $c \in \mathbb{R}$
4) The limit $\lim_{n\to\infty} \frac{\|\mathbf{z}\|_2}{\sqrt{n}} \in [0, \infty)$ exists
5) For any iteration $t \in \mathbb{N}$ and variance $\sigma^2 \geq 0$, the limit $\lim_{qL\to\infty} \frac{1}{qL}\mathbb{E}\left[\langle \mathbf{s}, \boldsymbol{\eta}_t(\mathbf{s} + \mathbf{Z})\rangle\right]$ exists and is finite for $\mathbf{Z} \sim \mathcal{N}\left(0, \sigma^2\mathbf{I}\right)$
6) For $s, t \in \mathbb{N}$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{2\times 2}$, $\lim_{qL\to\infty} \frac{1}{qL}\mathbb{E}\left[\langle \boldsymbol{\eta}_s(\mathbf{s} + \mathbf{Z}), \boldsymbol{\eta}_t(\mathbf{s} + \mathbf{Z}')\rangle\right]$ exists and is finite for $(\mathbf{Z}, \mathbf{Z}') \sim \mathcal{N}(0, \boldsymbol{\Sigma} \otimes \mathbf{I})$.

Then, the limits in the following state evolution recursion exist:

$$\begin{aligned} \tau_0^2 &= \lim_{n\to\infty} \frac{\|\mathbf{y}\|_2^2}{n} \\ \tau_{t+1}^2 &= \sigma^2 + \lim_{n\to\infty} \frac{1}{n}\mathbb{E}\left[\left\|\boldsymbol{\eta}_t\left(\mathbf{s} + \tau_t\boldsymbol{\zeta}_t\right) - \mathbf{s}\right\|_2^2\right], \end{aligned} \tag{9}$$

where $\boldsymbol{\zeta} \sim \mathcal{N}(0, \mathbf{I})$. If furthermore,

$$\frac{1}{n} \operatorname{div} \boldsymbol{\eta}_{t-1}\left(\mathbf{r}^{(t-1)}\right) \overset{\mathrm{P}}{=} \mathbb{E}\left[\frac{1}{n} \operatorname{div} \boldsymbol{\eta}_{t-1}\left(\mathbf{s} + \tau_{t-1}\boldsymbol{\zeta}\right)\right], \tag{10}$$

with $\tau_s > \sigma$ for all $s \leq t$, then, for any uniformly pseudo-Lipschitz function $\psi_n : (\mathbb{R}^{qL})^2 \to \mathbb{R}$ of order k and $qL \geq 1$, it follows that

$$\psi_n\left(\mathbf{s}^{(t)} + \mathbf{A}^{\mathrm{T}}\mathbf{z}^{(t)}, \mathbf{s}\right) \overset{\mathrm{P}}{=} \mathbb{E}\left[\psi_n\left(\mathbf{s} + \tau_t\boldsymbol{\zeta}_t, \mathbf{s}\right)\right], \tag{11}$$

where $\boldsymbol{\zeta}_t \sim \mathcal{N}(0, \mathbf{I})$ $\forall t$.

Corollary 3 (Adapted from [35]): Consider the Lipschitz-continuous MSE function $\psi(\mathbf{x}, \mathbf{y}) = \frac{1}{n}\|\boldsymbol{\eta}(\mathbf{x}) - \mathbf{y}\|_2^2$. Under

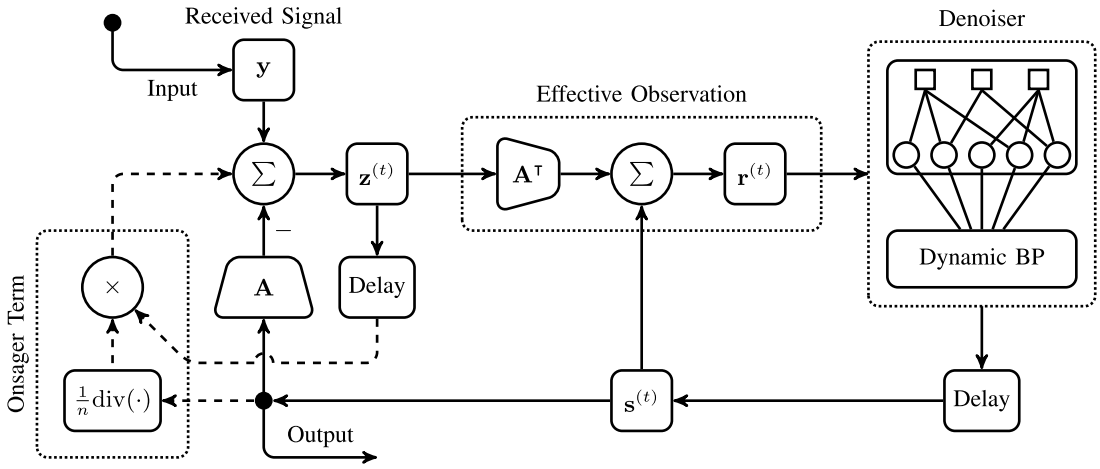

Fig. 2. Graphical representation of the AMP-BP algorithm, which seeks to recover $\mathbf{s}$ given observation $\mathbf{y}$, sensing matrix $\mathbf{A}$, and the outer LDPC code.

Theorem 2, it follows that

$$\begin{aligned}\frac{1}{n}\|\mathbf{s}^{(t+1)}-\mathbf{s}\|_2^2 &= \frac{1}{n}\|\boldsymbol{\eta}_t(\mathbf{s}^{(t)}+\mathbf{A}^{\mathrm{T}}\mathbf{z}^{(t)})-\mathbf{s}\|_2^2 \\ &\overset{P}{=} \frac{1}{n}\mathbb{E}\left[\|\boldsymbol{\eta}_t(\mathbf{s}+\tau_t\boldsymbol{\zeta}_t)-\mathbf{s}\|_2^2\right] \\ &= \tau_{t+1}^2-\sigma^2. \end{aligned} \tag{12}$$

Thus, the state evolution iterate may be used to track the expected MSE of the state estimates of AMP.

Corollary 4: Under Theorem 2, the effective observation $\mathbf{r}^{(t)}$ is asymptotically distributed as $\mathbf{s}+\tau_t\boldsymbol{\zeta}_t$, where $\boldsymbol{\zeta}_t \sim \mathcal{N}(0,\mathbf{I})$ and τ_t is a deterministic scalar.

Corollary 4 shows that, during each AMP iteration, the effective observation is asymptotically distributed as the true state embedded in Gaussian noise. This fact serves as the basis for the proposed denoising function, which is described in detail in the next section. After describing the denoiser, we show that the requirements of Theorem 2 are satisfied for the proposed SR-LDPC decoding algorithm.

III. BP Denoiser

In this section, we introduce the denoising function we wish to employ within AMP. To begin, we emphasize that $\mathbf{r}$ admits a sectionized representation akin to that of the state vector $\mathbf{s}$ in (4). That is, we can view both the state estimate and the effective observation as a concatenation of L vectors, each of length q. Mathematically, we have

$$\mathbf{r}=\begin{bmatrix}\mathbf{r}_1\\ \vdots \\ \mathbf{r}_L\end{bmatrix} \qquad \hat{\mathbf{s}}=\begin{bmatrix}\hat{\mathbf{s}}_1\\ \vdots \\ \hat{\mathbf{s}}_L\end{bmatrix}.$$

Moving forward, we temporarily neglect the superscript (t), which denotes the iteration count, to lighten notation; instead, we employ the hat symbol to distinguish the estimate $\hat{\mathbf{s}}$ from the true state vector $\mathbf{s}$. Fig. 2 illustrates several of the key quantities we employ throughout.

Consider the effective observation restricted to section ℓ. Under Corollary 4, the distribution of random observation vector $\mathbf{R}_\ell$ given section $\mathbf{S}_\ell=\mathbf{e}_g$ is given by

$$f_{\mathbf{R}_\ell|\mathbf{S}_\ell}(\mathbf{r}_\ell|\mathbf{e}_g)=\frac{1}{(2\pi)^{\frac{q}{2}}\tau^q}\exp\left(-\frac{\|\mathbf{r}_\ell-\mathbf{e}_g\|^2}{2\tau^2}\right). \tag{13}$$

It may be beneficial to think of the inner AMP loop as being equivalent to accessing a Gaussian vector channel L times, with every channel use being attached to one LDPC symbol in a manner akin to pulse position modulation (PPM). Under a uniform prior on $\mathbf{S}_\ell$, the conditional distribution of $\mathbf{S}_\ell$, and by extension, the corresponding coded symbol V_ℓ, becomes

$$\begin{aligned}\boldsymbol{\alpha}_\ell(g) &= \Pr(\mathbf{S}_\ell=\mathbf{e}_g|\mathbf{R}_\ell=\mathbf{r}_\ell) \\ &= \Pr(V_\ell=g|\mathbf{R}_\ell=\mathbf{r}_\ell)=\frac{f_{\mathbf{R}_\ell|V_\ell}(\mathbf{r}_\ell|g)}{\sum_{h\in\mathbb{F}_q}f_{\mathbf{R}_\ell|V_\ell}(\mathbf{r}_\ell|h)} \\ &= \frac{e^{-\frac{\|\mathbf{r}_\ell-\mathbf{e}_g\|^2}{2\tau^2}}}{\sum_{h\in\mathbb{F}_q}e^{-\frac{\|\mathbf{r}_\ell-\mathbf{e}_h\|^2}{2\tau^2}}}=\frac{e^{\frac{\mathbf{r}_\ell(g)}{\tau^2}}}{\sum_{h\in\mathbb{F}_q}e^{\frac{\mathbf{r}_\ell(h)}{\tau^2}}}. \end{aligned} \tag{14}$$

A possible estimate for $\mathbf{S}_\ell$ can be formed by taking its conditional expectation, given observation $\mathbf{R}_\ell=\mathbf{r}_\ell$, with

$$\mathbb{E}[\mathbf{S}_\ell|\mathbf{R}_\ell=\mathbf{r}_\ell]=\sum_{g\in\mathbb{F}_q}\mathbf{e}_g\Pr(\mathbf{S}_\ell=\mathbf{e}_g|\mathbf{R}_\ell=\mathbf{r}_\ell)=\boldsymbol{\alpha}_\ell. \tag{15}$$

A variant of this approach can be found in [24] for a system without an outer code. It is also employed in [29] in the context of unsourced random access. Yet, this approach overlooks the redundancy found in the outer code for the system under consideration. Ideally, we would like to take advantage of the outer code with the more precise MMSE estimate of the form

$$\mathbb{E}[\mathbf{S}_\ell|\mathbf{R}=\mathbf{r}]=\sum_{\mathbf{v}\in\mathcal{V}}\frac{\mathbf{e}_{v_\ell}\prod_{i=1}^{L}\Pr(\mathbf{R}_i=\mathbf{r}_i|\mathbf{S}_i=\mathbf{e}_{v_i})}{\sum_{\mathbf{v}'\in\mathcal{V}}\prod_{j=1}^{L}\Pr\left(\mathbf{R}_j=\mathbf{r}_j|\mathbf{S}_j=\mathbf{e}_{v'_j}\right)},$$

where $\mathcal{V}$ denotes the set of valid non-binary LDPC codes. However, as $|\mathcal{V}|$ scales exponentially in B, the MMSE denoiser quickly becomes computationally intractable. A viable alternative that trades off performance and complexity is to perform BP on the factor graph of the outer LDPC code. Implicitly, this approach computes an estimate for every $\mathbf{S}_\ell$ based on the observations contained within the corresponding computation tree of the code, up to a certain depth [8], [38].

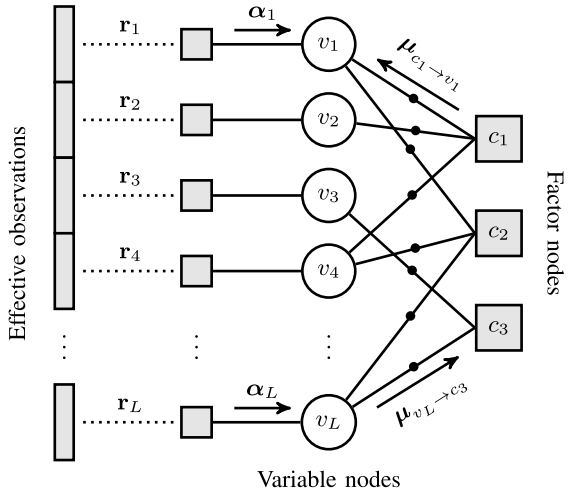

Fig. 3. Each section of the effective observation represents a single coded symbol; thus, the sections are linked together via the factor graph of the outer LDPC code.

Frameworks for performing BP on factor graphs are well-established [39]; thus, we assume some familiarity with such iterative procedures. For the q-ary LDPC portion of the article, we borrow definitions and concepts from Bennatan and Burshtein [7], who offer a comprehensive exposition of q-ary LDPC codes. We proceed by first considering the nuances of non-binary LDPC factor graphs, then presenting a BP algorithm, then proposing a dynamic BP denoiser for SR-LDPC codes, and finally by considering the properties of the proposed denoiser.

A. Non-Binary LDPC Graphs

The factor graph for an $\mathbb{F}_q$ LDPC code features L variable (left) nodes, which correspond to the symbols of the codewords, and $L(1-R)$ check (right) nodes enforcing parity constraints, where R is the *design rate* of the LDPC code [6], [7], [8]. An important distinction between binary and non-binary LDPC codes is that a factor graph for a non-binary LDPC code typically includes edge labels, which take values in $\mathbb{F}_q \setminus \{0\}$. Fig. 3 offers a notional factor graph for a non-binary LDPC code, where the edge labels are represented as dots along the graph edges. A vector $\mathbf{v} \in \mathbb{F}_q^L$ is a valid codeword if it satisfies the parity equations

$$\sum_{v_\ell \in N(c_p)} \omega_{\ell,p} \otimes v_\ell = 0 \qquad \forall p \in [L(1-R)], \tag{16}$$

where $N(c_p)$ is the collection of variable nodes adjacent to parity check c_p, i.e., neighbors on the factor graph. The summation and the multiplication operator $\otimes$ in (16) take place over finite field $\mathbb{F}_q$. Parameter $\omega_{\ell,p}$ represents the label or weight assigned with the edge connecting variable node v_ℓ and check node c_p. Adopting common factor graph concepts [39], [40], we denote the graph neighbors of variable node v_ℓ by $N(v_\ell)$. The factor associated with c_p and derived from parity equation (16) can be expressed as an indicator function

$$\mathcal{G}_p(\mathbf{v}_p) = \mathbf{1}\left(\sum_{v_\ell \in N(c_p)} \omega_{\ell,p} \otimes v_\ell = 0\right), \tag{17}$$

where $\mathbf{v}_p = (v_\ell \in N(c_p))$ is a shorthand notation for the restriction of $\mathbf{v}$ to entries associated with graph neighborhood $N(c_p)$. With these definitions, the factor function associated with our LDPC code assumes the product decomposition given by

$$\mathcal{G}(\mathbf{v}) = \prod_{p \in [L(1-R)]} \mathcal{G}_p(\mathbf{v}_p). \tag{18}$$

Succinctly, $\mathcal{G}(\mathbf{v})$ is an indicator function that assesses whether its argument is a valid codeword.

To create a suitable LDPC code, one can first construct a Tanner graph [41] according to established techniques [8], [42] and then assign labels to edges, possibly randomly and independently from a uniform distribution over $\mathbb{F}_q \setminus \{0\}$.

B. Belief Propagation

We view messages for a non-binary LDPC code as multi-dimensional belief vectors over $\mathbb{F}_q$. Messages from variable nodes to check nodes are denoted as $\boldsymbol{\mu}_{v \to c}$, and messages in the reverse direction are represented as $\boldsymbol{\mu}_{c \to v}$. Formally, a message going from check node c_p to variable node $v_\ell \in N(c_p)$ is computed component-wise through the equation

$$\boldsymbol{\mu}_{c_p \to v_\ell}(g) = \sum_{\mathbf{v}_p : v_\ell = g} \mathcal{G}_p(\mathbf{v}_p) \prod_{v_j \in N(c_p) \setminus v_\ell} \boldsymbol{\mu}_{v_j \to c_p}(g_j). \tag{19}$$

While (19) is shown in compact form, the actual summation operation is cumbersome. Finding the set of summands entails identifying sequences of the form $(g_j \in \mathbb{F}_q : v_j \in N(c_p) \setminus v_\ell)$ that fulfill local condition (16) or, equivalently,

$$\sum_{v_j \in N(c_p) \setminus v_\ell} \omega_{j,p} \otimes g_j = -\omega_{\ell,p} \otimes g. \tag{20}$$

Likewise, a belief vector passed from variable node v_ℓ to check node c_p, where $p \in N(v_\ell)$, is calculated component-wise via

$$\boldsymbol{\mu}_{v_\ell \to c_p}(g) \propto \boldsymbol{\alpha}_\ell(g) \prod_{c_\xi \in N(v_\ell) \setminus c_p} \boldsymbol{\mu}_{c_\xi \to v_\ell}(g). \tag{21}$$

The '$\propto$' symbol indicates that the positive measure should be normalized before being sent out as a message. Vector $\boldsymbol{\alpha}_\ell$ in (21) can be viewed as a collection of beliefs based on local observations, as in (14). That is, entry $\boldsymbol{\alpha}_\ell(g)$ captures the posterior probability that symbol g is the true field element within section ℓ, given the local observation. Other BP messages are initialized with $\boldsymbol{\mu}_{v \to c} = \mathbf{1}$ and $\boldsymbol{\mu}_{c \to v} = \mathbf{1}$. These message passing operations appear in Fig. 3. The traditional parallel sum-product algorithm iterates between (19) and (21), alternating between updated rightbound messages and leftbound messages.

One of the key advantages of indexing vectors using field elements in $\mathbb{F}_q$, as pointed out in [7], is the ensuing ability to define pertinent operators on these vectors. Paralleling existing literature, we consider two operators.

Definition 5 (Vector $+g$ Operator [7]): For field element $g \in \mathbb{F}_q$, the vector $+g$ operator acting on $\mathbf{b} \in \mathbb{R}^q$ and denoted by $\mathbf{b}^{+g}$ is defined as

$$\begin{aligned}\mathbf{b}^{+g} &= \left(b_g, b_{g\oplus 1}, \ldots, b_{g\oplus(q-1)}\right) \\ &= \left(b_{h\oplus g} : h \in \mathbb{F}_q\right),\end{aligned}$$

where subscript addition $\oplus$ is performed in $\mathbb{F}_q$.

Definition 6 (Vector $\times g$ Operator [7]): For field element $g \in \mathbb{F}_g \setminus \{0\}$, we define the vector $\times g$ operator acting on $\mathbf{b} \in \mathbb{R}^q$ and denoted by $\mathbf{b}^{\times g}$ by

$$\begin{aligned}\mathbf{b}^{\times g} &= \left(b_0, b_g, b_{2\otimes g}, \ldots, b_{(q-1)\otimes g}\right) \\ &= \left(b_{h\otimes g} : h \in \mathbb{F}_q\right)\end{aligned}$$

where subscript product $\otimes$ takes place in $\mathbb{F}_q$.

We emphasize that the $+g$ and $\times g$ operators introduced above are reversible, with

$$\begin{aligned}\left(\mathbf{b}^{+g}\right)^{-g} &= \mathbf{b}g \quad \in \mathbb{F}_q \\ \left(\mathbf{b}^{\times g}\right)^{\times g^{-1}} &= \mathbf{b}g \quad \in \mathbb{F}_q \setminus \{0\}.\end{aligned}$$

These operations essentially permute the entries of $\mathbf{b}$ in a structured fashion that naturally meshes with field actions. These operations are especially meaningful in the computation of BP messages for non-binary LDPC codes, as factor nodes impose constraints that are easily expressible within the Galois field $\mathbb{F}_q$. Vector operators then become a convenient way to track the distribution of belief vectors during message passing. Specifically, under these operations, we can rewrite (19) as

$$\boldsymbol{\mu}_{c_p\to v_\ell} = \left(\bigodot_{v_j\in N(c_p)\setminus v_\ell} \left(\boldsymbol{\mu}_{v_j\to c_p}\right)^{\times\omega_{j,p}^{-1}}\right)^{\times(-\omega_{\ell,p})}, \tag{22}$$

where $\omega_{j,p}$ is the label on the edge between variable node v_j and factor node c_p [7]. Here, the operator $\odot$ denotes the $\mathbb{F}_q$-convolution between two vectors,

$$\left[\boldsymbol{\mu}\odot\boldsymbol{\nu}\right]_g = \sum_{h\in\mathbb{F}_q} \mu_h \cdot \nu_{g-h} \qquad g \in \mathbb{F}_q.$$

The exposition can be simplified further if we absorb the edge labels within the messages themselves. Specifically, we adopt the definitions

$$\overline{\boldsymbol{\mu}}_{v_j\to c_p} = \left(\boldsymbol{\mu}_{v_j\to c_p}\right)^{\times\omega_{j,p}^{-1}} \tag{23}$$

$$\overline{\boldsymbol{\mu}}_{c_p\to v_\ell} = \left(\boldsymbol{\mu}_{c_p\to v_\ell}\right)^{\times\left(-\omega_{\ell,p}^{-1}\right)}. \tag{24}$$

Then, (22) morphs into the simpler expression

$$\overline{\boldsymbol{\mu}}_{c_p\to v_\ell} = \bigodot_{v_j\in N(c_p)\setminus v_\ell} \overline{\boldsymbol{\mu}}_{v_j\to c_p}. \tag{25}$$

This equation highlights the role of the $\mathbb{F}_q$-convolution within BP for non-binary LDPC codes. The message from variable node v_ℓ to check node c_p found in (21) also admits a more compact form. For $p \in N(v_\ell)$, the traditional outgoing message from a variable node can be written as

$$\boldsymbol{\mu}_{v_\ell\to c_p} = \frac{\boldsymbol{\alpha}_\ell \circ \left(\bigcirc_{c_\xi\in N(v_\ell)\setminus c_p} \boldsymbol{\mu}_{c_\xi\to v_\ell}\right)}{\left\|\boldsymbol{\alpha}_\ell \circ \left(\bigcirc_{c_\xi\in N(v_\ell)\setminus c_p} \boldsymbol{\mu}_{c_\xi\to v_\ell}\right)\right\|_1}, \tag{26}$$

where $\circ$ denotes the Hadamard product between two vectors, and $\bigcirc$ represents the Hadamard product of a collection of vectors.

A natural estimate for the distribution associated with variable node v_ℓ, including intrinsic information, is

$$\hat{\mathbf{S}}_\ell = \frac{\boldsymbol{\alpha}_\ell \circ \left(\bigcirc_{c_p\in N(v_\ell)} \boldsymbol{\mu}_{c_p\to v_\ell}\right)}{\left\|\boldsymbol{\alpha}_\ell \circ \left(\bigcirc_{c_p\in N(v_\ell)} \boldsymbol{\mu}_{c_p\to v_\ell}\right)\right\|_1}. \tag{27}$$

As we will see shortly, (27) is the output of our proposed denoiser.

Remark 7: In our construction, $q = 2^m, m \geq 1$ because indexing is derived from sequences of bits. This invites the application of fast techniques to implement message passing over the corresponding factor graph. Specifically, the fast Walsh-Hadamard transform (FWHT) can be utilized to rapidly and efficiently compute (25), with

$$\overline{\boldsymbol{\mu}}_{c_p\to v_\ell} \propto \text{fwht}^{-1}\left(\prod_{v_j\in N(c_p)\setminus v_\ell} \text{fwht}\left(\overline{\boldsymbol{\mu}}_{v_j\to c_p}\right)\right).$$

This technique is especially meaningful given that it may be desirable to maintain large sections and, hence, a large alphabet size for sparse regression codewords. Alternatively, one could adopt a different finite field convolution or a ring structure amenable to the circular convolution to create local factor functions conducive to the fast Fourier transform [6], [43], [44].

With these tools in mind, we are ready to formally define our proposed denoiser.

C. BP Denoiser

Conceptually, one can initiate the state of the LDPC factor graph using the effective observation $\mathbf{r}$, run a few rounds of BP, and then form an estimate for the state based on (27). As mentioned before, in the absence of BP iterations, local estimates reduce to the conditional expectation $\mathbb{E}\left[\mathbf{S}_\ell|\mathbf{R}_\ell = \mathbf{r}_\ell\right]$ found in (15). Yet, as more iterations of the BP algorithm are performed, the estimate for $\mathbf{S}_\ell$ can be refined based on the computation tree of the outer code, up to a certain depth.

Definition 8 (BP Denoiser): Let N_t denote the number of BP iterations to perform during AMP iteration t. The BP denoiser:

1) initializes the LDPC factor graph with estimates $\boldsymbol{\alpha}_\ell$ computed from $\mathbf{r}_\ell$ for $\ell \in [L]$ according to (14);
2) computes and passes variable to check messages (see (26)) and check to variable messages (see (23), (25), (24) and Remark 7) along the edges of the factor graph in an alternating fashion N_t times;
3) computes updated state estimates according to (27).

The output of this denoiser can then be passed to the AMP composite algorithm for the computation of the next residual, enhanced with the Onsager term.

To the reader familiar with the iterative decoding of LDPC codes, it may seem more natural to construct an estimate for the distribution associated with variable node v_ℓ based on

extrinsic information, i.e.,

$$\hat{\mathbf{S}}_\ell^{\text{ext}} = \frac{\bigcirc_{c_p \in N(v_\ell)} \boldsymbol{\mu}_{c_p \to v_\ell}}{\left\| \bigcirc_{c_p \in N(v_\ell)} \boldsymbol{\mu}_{c_p \to v_\ell} \right\|_1}. \tag{28}$$

One may be tempted to argue that $\boldsymbol{\alpha}_\ell$ should not be used when passing a message back to the left-most nodes in Fig. 3. However, the presence of the Onsager term in (6) serves to break first-order dependencies and, hence, one need not worry about the presence of $\boldsymbol{\alpha}_\ell$ in (27) as part of the iterative process.

When AMP is used with the BP denoiser as presented in this paper, the algorithm is referred to as the *AMP-BP* algorithm.

D. Properties of BP Denoiser

The BP denoiser relies on the effective observation $\mathbf{r}$ being asymptotically distributed as the true state embedded in i.i.d. Gaussian noise, or

$$\mathbf{r}^{(t)} \sim \mathbf{s} + \tau_t \boldsymbol{\zeta}_t, \tag{29}$$

where τ_t is a deterministic quantity and $\boldsymbol{\zeta}_t$ has i.i.d. $\mathcal{N}(0,1)$ components. This assumption is valid under the conditions of Theorem 2; thus, having defined the BP denoiser, we now revisit these conditions to ensure that they are met. For the sake of conciseness, we will not provide rigorous proofs for most of these conditions; however, they may be proved under our construction of $\mathbf{A}$, the SNR constraint, and the properties of the AWGN. We do, however, provide a rigorous proof that the denoiser is uniformly Lipschitz under the following two conditions.

Condition 9 (Sub-Girth BP): The BP denoiser is said to possess the *Sub-Girth BP* condition when fewer message passing iterations are performed on the factor graph of the LDPC code than the shortest cycle of this same graph, per AMP denoising step.

This condition is reasonable because, in contrast to the traditional technique of performing many BP iterations at once, we are primarily interested in repeatedly performing a few BP iterations at a time as the BP algorithm is run within each AMP iteration. Though this condition is sufficient for the theory to hold, in practice, one may be able to violate this condition and still obtain reasonable performance. With this condition in mind, we obtain the following result.

Condition 10: As the dimension of the block sparse vector $\mathbf{s} \in \mathbb{R}^{qL}$ is scaled to infinity, the following constraints are satisfied:

1) The left and right degrees of the outer LDPC code do not exceed $d_{\max}^v < \infty$ and $d_{\max}^c < \infty$, respectively.
2) The number of BP iterations performed does not exceed $N_{\max} < \infty$, regardless of the girth of the factor graph.
3) The field size q of the outer LDPC code does not exceed $q_{\max} < \infty$.

Note that under Condition 10, it is still possible to let $qL \to \infty$ by scaling $L \to \infty$. Under these two conditions, we obtain the following result.

Theorem 11 (BP Denoiser is Uniformly Lipschitz): Under Conditions 9 and 10, the BP denoiser presented in Definition 8 is uniformly Lipschitz continuous.

For the proof of this and all other results in this section, please see Appendix A.

Since the conditions of Theorem 2 are satisfied under Conditions 9 and 10, we conclude that state evolution holds for the AMP-BP algorithm. This endows the algorithm with a significant amount of mathematical structure that will be exploited in Section IV for performance prediction and optimization.

The final step in completing our AMP-BP algorithm is computing the Onsager correction term, which we provide in Proposition 12.

Proposition 12: The Onsager correction term associated with the BP denoiser is given by

$$\begin{aligned} &\frac{\mathbf{z}^{(t-1)}}{n} \operatorname{div} \boldsymbol{\eta}_{t-1}\left(\mathbf{r}^{(t-1)}\right) \\ &= \frac{\mathbf{z}^{(t-1)}}{n\tau^2}\left(\left\|\boldsymbol{\eta}_{t-1}\left(\mathbf{r}^{(t-1)}\right)\right\|_1 - \left\|\boldsymbol{\eta}_{t-1}\left(\mathbf{r}^{(t-1)}\right)\right\|_2^2\right). \end{aligned} \tag{30}$$

Note that this term has a particularly simple form that is amenable to efficient computation. In the next section, we consider how to use the state evolution algorithm for performance prediction and code optimization.

IV. State Evolution

Recall that state evolution is an iterative algorithm that seeks to characterize the performance of AMP as a function of its iteration count t. Under Corollary 4, the effective observation is distributed as the true state embedded in zero-mean i.i.d. Gaussian noise with variance τ_t^2. Using the state evolution formalism, the value of τ_t^2 at iteration t can be computed through the following recursion:

$$\begin{aligned} \tau_0^2 &= \lim_{n\to\infty} \frac{\|\mathbf{y}\|_2^2}{n} \\ \tau_{t+1}^2 &= \sigma^2 + \lim_{n\to\infty} \frac{1}{n}\mathbb{E}\left[\left\|\boldsymbol{\eta}_t\left(\mathbf{s} + \tau_t\boldsymbol{\zeta}_t\right) - \mathbf{s}\right\|_2^2\right], \end{aligned} \tag{31}$$

where n is the number of channel uses, σ^2 is the channel noise variance, and $\boldsymbol{\zeta}_t \sim \mathcal{N}(0, \mathbf{I})$. Here, the expectation is computed with a uniform distribution over vectors $\mathbf{s}$.

As shown in Corollary 3, the state evolution algorithm may be used to compute the expected mean-squared-error (MSE) of the state estimate at iteration t. Thus, by comparing the limiting values of τ_t^2 for various configurations of hyperparameters (e.g. number of channel uses n, LDPC code length L, field size q, etc), we can identify which set of parameters provides the lowest reconstruction MSE $\|\mathbf{s}^{(t)} - \mathbf{s}\|_2^2$.

However, naively computing $\lim_{t\to\infty} \tau_t^2$ is computationally intensive, in part due to the expectation which must be taken over the set of indexed $\mathbb{F}_q$ LDPC codewords. This is not ideal, as the SR-LDPC code designer may have a large parameter space to search over, and naively using the state evolution algorithm to find the optimal code configuration may take a prohibitively large amount of time. Thus, we are interested in finding a more efficient way of computing the state evolution for a given SR-LDPC code.

Note that $\mathbb{E}\left[\tau_0^2\right] = \sigma^2 + \frac{L}{n}$; thus τ_0^2 may be directly approximated from the system parameters without the need for any high-dimensional computations. Additionally, the expected

MSE of the input to the denoiser can be computed directly from τ_t^2 and the system parameters. If we could efficiently compute the expected MSE at the output of the denoiser given the expected MSE of its input, then we could also compute τ_{t+1}^2. Then, we would have a low-dimensional recursion for predicting the performance of a SR-LDPC code under a given hyperparameter set, thus enabling rapid performance prediction and code optimization.

The purpose of this section is to study the mathematical properties of SR-LDPC codewords and BP denoising in order to develop such a recursive algorithm. Though we do not provide an exact algorithm for computing the state evolution, we do provide an *approximate* state evolution algorithm in Definition 13, which is the main result of this section. This approximate solution is based on the idea of "MSE message passing", whereby scalar expected MSE values are passed along the edges of a bipartite graph representing the constraints of the outer LDPC code. Using the known system parameters as well as the current value of τ_t^2, the starting expected MSE of each section may be computed. Then, expected MSEs are passed between variable and check nodes, where nodes provide approximate expected output MSEs based on their inputs. After a fixed number of message passing iterations, the final output MSE may be computed, which is then used in the computation of τ_{t+1}^2.

We begin this section by introducing the approximate state evolution algorithm in Definition 13. Then, we explore the properties of SR-LDPC codes that enable the development of the algorithm, and finally we evaluate the algorithm's performance and utility.

Definition 13 (Approximate State Evolution): Let T denote the number of AMP iterations to model, let $\{N_t : t \in [T]\}$ denote the number of BP rounds to perform during each AMP iteration $t \in [T]$, and let σ^2 denote the channel noise variance. The approximate state evolution algorithm proceeds by

1) Approximate $\tau_0^2 \approx \mathbb{E}\left[\tau_0^2\right] = \sigma^2 + \frac{L}{n}$
2) For each AMP iteration $t \in [T]$, consider a bipartite graph representing the constraints of the outer LDPC code where all edges have weight one. Then,
   a) Initialize all graph messages to be $\frac{1}{q}$
   b) Set $\boldsymbol{\mu}_{c_0 \to v_\ell} = \mathbb{E}\left[\boldsymbol{\alpha}_\ell(0)\right] \ \forall \ell \in [L]$
   c) For each BP iteration $\varsigma \in [N_t]$:
      i) Send variable to check node messages according to Proposition 34, where

$$\boldsymbol{\mu}_{v_\ell \to c_p} = 1 - \Psi\left(\frac{1}{\frac{1}{\tau_t^2} + \sum_{c_\xi \in N(v_\ell)\setminus c_p} \frac{1}{\tau_{\xi,\ell}^2}}\right),$$

      where $\tau_{\xi,\ell}^2 = \Psi^{-1}\left(\tilde{\boldsymbol{\mu}}_{c_\xi \to v_\ell}(0)\right)$ and $\Psi\left(\tau_t^2\right) = \mathbb{E}\left[\boldsymbol{\alpha}(0)\right]$.
      ii) Send check to variable node messages according to Proposition 33, where

$$\boldsymbol{\mu}_{c_p \to v_\ell} = \frac{q-1}{q} - \left(\frac{q}{q-1}\right)^{\gamma-1} \times \prod_{\substack{v_j \in N(c_p) \\ v_j \neq v_\ell}} \left(\mathbb{E}\left[\left\|\boldsymbol{\mu}_{v_j \to c_p}\right\|_2^2\right] - \frac{1}{q}\right),$$

      where $\gamma = |N(c_p)| - 1$.
   d) Approximate expected output MSE using Proposition 34, where

$$\mathbb{E}\left[\|\mathbf{S}_\ell - \hat{\mathbf{S}}_\ell\|_2^2\right] = 1 - \Psi\left(\frac{1}{\frac{1}{\tau_t^2} + \sum_{c_p \in N(v_\ell)} \frac{1}{\tau_{p,\ell}^2}}\right),$$

   e) Compute $\tau_{t+1}^2 = \sigma^2 + \frac{1}{n}\sum_{\ell \in [L]} \mathbb{E}\left[\|\mathbf{S}_\ell - \hat{\mathbf{S}}_\ell\|_2^2\right]$.

Having introduced the approximate state evolution recursion, we now study the properties of SR-LDPC codewords and BP denoising that enable the algorithm. We note that the proofs associated with all propositions, corollaries, lemmas, and theorems from this section are contained in Appendix B.

A. Geometric Uniformity of Indexed $\mathbb{F}_q$ LDPC Codes

We begin by examining the symmetry properties of indexed $\mathbb{F}_q$ LDPC codewords and show that SR-LDPC codewords are geometrically uniform. The notion of geometric uniformity, as presented in [45], is of great value because it guarantees that the error probability over a Gaussian channel does not depend on which codeword is transmitted. In particular, the sets of distances (distance profile) from any codeword to all other codewords are all the same.

First, note that the signal constellation produced by mapping a field element $g \in \mathbb{F}_q$ to vector element $\mathbf{e}_g \in \mathbb{R}^q$ is invariant under coordinate permutations. That is, suppose $\Pi : \mathbb{R}^q \to \mathbb{R}^q$ is a permutation matrix. then the following set equality (trivially) holds,

$$\{\mathbf{e}_g : g \in \mathbb{F}_q\} = \{\Pi \mathbf{e}_g : g \in \mathbb{F}_q\}. \tag{32}$$

Furthermore, it is known that every coordinate permutation operator is an *isometry*, with

$$\|\Pi \mathbf{e}_g - \Pi \mathbf{e}_h\|^2 = \|\mathbf{e}_g - \mathbf{e}_h\|^2 \quad \forall g, h \in \mathbb{F}_q, \tag{33}$$

where $\pi : [q] \to [q]$ is the permutation function corresponding to matrix Π. It follows that the sets in (32) are *geometrically congruent* under any permutation operator.

We can extend these observations to state vectors in $\mathbb{R}^{qL}$. Consider a set of permutation matrices on $\mathbb{R}^{q \times q}$, which we denote by $\Pi^{(1)}, \ldots, \Pi^{(L)}$. Define the block diagonal permutation matrix $\mathbf{\Pi} = \operatorname{diag}\left(\Pi^{(1)}, \ldots, \Pi^{(L)}\right)$. Given that we can write

$$\|\mathbf{s} - \mathbf{s}'\|^2 = \sum_{\ell=1}^{L} \|\mathbf{s}_\ell - \mathbf{s}'_\ell\|^2,$$

we deduce that the original codebook $\mathcal{S} \subset \mathbb{R}^{qL}$ and any section-wise permutation $\mathbf{\Pi}$ thereof must also be *geometrically congruent*.

Definition 14 (Geometric Uniformity [45]): A signal set $\mathcal{S}$ is *geometrically uniform* if, given any two points $\mathbf{s}$ and $\mathbf{s}'$ in $\mathcal{S}$, there exists an isometry that transforms $\mathbf{s}$ to $\mathbf{s}'$ while leaving $\mathcal{S}$ invariant.

Since any section-wise permutation $\mathbf{\Pi}$ acting on $\mathcal{S}$ produces a symmetry of $\mathcal{S}$, it becomes straightforward to show that this set is geometrically uniform.

Proposition 15: Let $\mathcal{S}$ be the codebook produced by combining the $\mathbb{F}_q$ LDPC outer code and the indexing step. Then, the set $\mathcal{S}$ is geometrically uniform.

This result should not be too surprising to the reader familiar with LDPC codes, vector indexing, and sparse regression codes. Nevertheless, this is important because it permits an analysis of the system under the all-zero codeword. We elaborate on the section symmetry in the following proposition.

Proposition 16: Let Π be any permutation on the entries of vectors in $\mathbb{R}^q$. The distribution of $\mathbf{R}_\ell$ conditioned on the input $\mathbf{S}_\ell$ is permutation invariant in the sense that

$$f_{\mathbf{R}_\ell|\mathbf{S}_\ell}\left(\mathbf{r}_\ell|\mathbf{e}_g\right) = f_{\mathbf{R}_\ell|\mathbf{S}_\ell}\left(\Pi\mathbf{r}_\ell\middle|\mathbf{e}_{\pi(g)}\right) \tag{34}$$

for any $\mathbf{r}_\ell \in \mathbb{R}^q$. Above, $\pi(\cdot)$ is a representation of Π where $\pi(g)$ denotes the permutation of the integer position of g under the bijection of Remark 1. In other words, $\mathbf{e}_{\pi(g)} = \Pi\mathbf{e}_g$ for any $g \in \mathbb{F}_q$.

While there are only q permutations induced through field mapping of the form $g \mapsto g \oplus u$, determined by choosing $u \in \mathbb{F}_q$, the mathematical statement holds for all $q!$ possible permutations of the vector indices. Thus, this attribute forms a strong notion of statistical symmetry that is related to a symmetry property of binary LDPC codes [46] and non-binary LDPC codes over finite fields [7]. As mentioned in the latter article, the capacity-achieving distribution for constrained channels with such statistical symmetry is uniform over the q possible inputs. Fortunately, the marginal input distribution to the Gaussian vector channel corresponding to section ℓ under SR-LDPC encoding is indeed uniform over the admissible inputs.

Corollary 17: Suppose that vector $\mathbf{e}_0$ is the input to the Gaussian vector channel of Corollary 4. Then, observation vectors $\mathbf{R}_\ell$ and $\mathbf{R}_\ell^{\times\omega}$, where $\omega \in \mathbb{F}_q \setminus \{0\}$, have identical distributions.

This corollary is pertinent because, as discussed above, system analysis for a geometrically uniform codebook can be performed assuming that the all-zero codeword has been transmitted. Under such circumstances, the action of edge label $\omega_{\ell,p}$ does not affect the distribution of the rightbound messages $\boldsymbol{\mu}_{v_\ell \to c_p}$. This complexity reduction also extends to the distribution of $\boldsymbol{\mu}_{c_p \to v_\ell}$. Hence, the effects of the edge labels can be disregarded when studying the statistical properties of the BP denoiser.

B. Statistical Properties of BP Messages

We now explore certain statistical properties of the BP messages that are passed during SR-LDPC decoding. Throughout the remainder of this section, we assume that the all-zero codeword has been sent (i.e., $v_\ell = 0\ \forall \ell \in [L]$) and we assume that Corollary 4 holds. We begin by introducing the notions of likelihood-vectors and likelihood-vector random variables.

Definition 18: A *likelihood-vector* $\boldsymbol{\ell} \in \mathbb{R}^q$ is a vector whose entries are non-negative, i.e. $\boldsymbol{\ell}(i) \geq 0\ \forall i \in [q]$.

Definition 19: A *likelihood-vector random variable* is defined as a random vector $\mathbf{L} = (L_0, L_1, \ldots, L_{q-1})$ in $\mathbb{R}^q$ that takes on values from the set of likelihood vectors. Furthermore, a likelihood-vector random variable $\mathbf{L}$ is called *group-symmetric* if

$$f_{\mathbf{L}}\left(\boldsymbol{\ell}\right) = f_{\mathbf{L}}\left(\boldsymbol{\ell}^{\times g}\right) \tag{35}$$

for any field element $g \in \mathbb{F}_q \setminus \{0\}$. Similarly, $\mathbf{L}$ is said to be *permutation-symmetric* if

$$f_{\mathbf{L}}\left(\boldsymbol{\ell}\right) = f_{\mathbf{L}}\left(\Pi_0\boldsymbol{\ell}\right) \tag{36}$$

for any permutation matrix Π_0 that preserves the location of the zeroth entry in its argument. Such a random vector is qualified as *dominant* if, in addition to *symmetry*, the mean of the zeroth element $\mathbb{E}[L_0]$ is greater than or equal to the expected value of any other entry.

Remark 20: We emphasize that, if a likelihood-vector random variable is either group-symmetric or permutation-symmetric, then the expected value of all its components, except for the zeroth entry, are equal. We can therefore unambiguously adopt the uniform notation $\mathbb{E}[L_\bullet]$, where $\bullet$ can be any field element $g \in \mathbb{F}_q \setminus \{0\}$.

A likelihood-vector random variable that takes on values in $\mathbb{R}^q$ can be normalized to produce a probability-vector random variable on $\mathbb{F}_q$. Such vector random variables are defined below. Both notions are important in analyzing the performance of sparse regression LDPC codes.

Definition 21: A *probability-vector random variable* is a random vector $\mathbf{D} = (D_0, D_1 \ldots, D_{q-1})$ in $\mathbb{R}^q$ that takes on values in the probability simplex. Such a probability-vector random variable $\mathbf{D}$ is called *group-symmetric* if

$$f_{\mathbf{D}}\left(\mathbf{d}\right) = f_{\mathbf{D}}\left(\mathbf{d}^{\times g}\right) \tag{37}$$

for any field element $g \in \mathbb{F}_q \setminus \{0\}$. Moreover, $\mathbf{D}$ is said to be *permutation-symmetric* if

$$f_{\mathbf{D}}\left(\mathbf{d}\right) = f_{\mathbf{D}}\left(\Pi_0\mathbf{d}\right) \tag{38}$$

for any permutation matrix Π_0 that preserves the location of the zeroth entry in its argument. Such a random vector is *dominant* if, in addition to *symmetry*, the mean of the zeroth element $\mathbb{E}[D_0]$ is greater than or equal to the expected value of any other entry.

The most important probability-vector random variables for the problem at hand are normalized likelihood-vector random variables. We adopt the notation

$$\bar{\mathbf{L}} = \frac{\mathbf{L}}{\|\mathbf{L}\|_1} \tag{39}$$

for the normalized version of a likelihood-vector random variable. Under Corollary 4, the components of likelihood vectors associated with the effective observation are derived from the Gaussian distribution. This fact, which we use extensively throughout, acts as a motivation for the next definition.

Definition 22: We define a *permutation-symmetric Gaussian likelihood-vector random variable* as a dominant permutation-symmetric likelihood-vector random variable that is component-wise equal to

$$L_g = f_{\mathbf{R}_\ell|\mathbf{S}_\ell}\left(\mathbf{R}_\ell|\mathbf{e}_g\right) \tag{40}$$

where $\mathbf{R}_\ell$ has distribution $f_{\mathbf{R}_\ell|\mathbf{S}_\ell}(\cdot|\mathbf{e}_0)$, as defined in (13). When a permutation-symmetric Gaussian likelihood-vector random variable $\mathbf{L}$ is normalized, we call the resulting vector $\bar{\mathbf{L}}$ a *permutation-symmetric Gaussian probability-vector random variable*.

Part of the motivation for introducing these definitions is rooted in the operations that take place on the factor graph of the $\mathbb{F}_q$ LDPC outer code during belief propagation. One benefit of working with the likelihood-vector, as opposed to the probability vector, is the fact that vector components in (40) are independent, with the joint distribution assuming a product form. We turn to the effects of the $\mathbb{F}_q$-convolution on random likelihood vectors and show that this operation preserves certain properties.

Lemma 23: The $\mathbb{F}_q$-convolution of a finite set of independent dominant group-symmetric likelihood-vector random variables produces a dominant group-symmetric likelihood-vector random variable.

Another interesting property of the $\mathbb{F}_q$-convolution of likelihood vectors pertains to the one-norm of the output. This result is analogous to the one-norm relation for the regular convolution; it is included below for the sake of completeness.

Lemma 24: Let $\{\boldsymbol{\ell}^{(p)}\}$ be a collection of likelihood vectors and define

$$\mathbf{n} = \bigodot_{p\in[\gamma]} \boldsymbol{\ell}^{(p)},$$

where γ is a natural number. Then, the one-norm of the output of the $\mathbb{F}_q$-convolution is equal to the product of the one-norms of the input vectors,

$$\|\mathbf{n}\|_1 = \prod_{p\in[\gamma]} \left\|\boldsymbol{\ell}^{(p)}\right\|_1.$$

Corollary 25: Let $\{\mathbf{L}^{(p)}\}$ be a collection of independent likelihood-vector random variables and define

$$\mathbf{N} = \bigodot_{p\in[\gamma]} \mathbf{L}^{(p)},$$

where γ is a natural number. Then, it necessarily holds that

$$\mathbb{E}\left[\|\mathbf{N}\|_1\right] = \prod_{p\in[\gamma]} \mathbb{E}\left[\left\|\mathbf{L}^{(p)}\right\|_1\right].$$

With the properties identified above, we can characterize the expectation of the $\mathbb{F}_q$-convolution of certain collections of likelihood-vector random variables. This is meaningful in that we can then track the mean behavior of certain BP messages passed on the factor graph of the outer LDPC code.

Proposition 26: Suppose $\{\mathbf{L}^{(p)}\}$ forms a collection of independent dominant group-symmetric likelihood-vector random variables. For any natural number γ, the expectation of $\mathbf{N} = \bigodot_{p\in[\gamma]} \mathbf{L}^{(p)}$ is governed by

$$\mathbb{E}[N_0] = \frac{1}{q}\prod_{p\in[\gamma]}\left(\mathbb{E}\left[L_0^{(p)}\right] + (q-1)\mathbb{E}\left[L_\bullet^{(p)}\right]\right) + \left(1-\frac{1}{q}\right)\prod_{p\in[\gamma]}\left(\mathbb{E}\left[L_0^{(p)}\right] - \mathbb{E}\left[L_\bullet^{(p)}\right]\right) \quad (41)$$

$$\mathbb{E}[N_\bullet] = \frac{1}{q}\prod_{p\in[\gamma]}\left(\mathbb{E}\left[L_0^{(p)}\right] + (q-1)\mathbb{E}\left[L_\bullet^{(p)}\right]\right) - \frac{1}{q}\prod_{p\in[\gamma]}\left(\mathbb{E}\left[L_0^{(p)}\right] - \mathbb{E}\left[L_\bullet^{(p)}\right]\right). \quad (42)$$

We can extend these findings to probability vectors of the form $\bar{\mathbf{L}} = \mathbf{L}/\|\mathbf{L}\|_1$, where $\mathbf{L}$ is a dominant group-symmetric likelihood-vector random variable.

Corollary 27: Suppose $\{\bar{\mathbf{L}}^{(p)}\}$ is a collection of independent dominant group-symmetric probability-vector random variables. Then, for any natural number γ, the expectation of $\bar{\mathbf{N}} = \bigodot_{p\in[\gamma]} \bar{\mathbf{L}}^{(p)}$ is governed by

$$\mathbb{E}\left[\bar{N}_0\right] = \frac{1}{q} + \left(\frac{q}{q-1}\right)^{\gamma-1}\prod_{p\in[\gamma]}\left(\mathbb{E}\left[\bar{L}_0^{(p)}\right] - \frac{1}{q}\right) \quad (43)$$

$$\mathbb{E}\left[\bar{N}_\bullet\right] = \frac{1}{q} - \frac{1}{q}\left(\frac{q}{q-1}\right)^{\gamma}\prod_{p\in[\gamma]}\left(\mathbb{E}\left[\bar{L}_0^{(p)}\right] - \frac{1}{q}\right). \quad (44)$$

An important application of Corollary 27 for our analysis is the situation where $\{\bar{\mathbf{L}}^{(p)}\}$ is a collection of independent permutation-symmetric Gaussian probability-vector random variables, each with parameter τ. Our last set of results on the statistical properties of likelihood vectors pertains to the two-norm of permutation-symmetric Gaussian probability-vector random variables.

Lemma 28: Suppose $\bar{\mathbf{L}}$ is a dominant permutation-symmetric Gaussian probability-vector random variable with standard deviation parameter τ. The expected two-norm of $\bar{\mathbf{L}}$ is related to $\mathbb{E}\left[\bar{L}_0\right]$ through the equation

$$\mathbb{E}\left[\left\|\bar{\mathbf{L}}\right\|_2^2\right] = \mathbb{E}\left[\bar{L}_0\right]. \quad (45)$$

Inspecting the proof of Lemma 28, one notices that (45) hinges on the property $\mathbb{E}\left[\bar{L}_\bullet^2\right] = \mathbb{E}\left[\bar{L}_0\bar{L}_\bullet\right]$. This relation arises naturally for dominant permutation-symmetric Gaussian probability-vector random variables, yet it may occur more generally. For instance, this property may be preserved under certain factor graph operations such as the $\mathbb{F}_q$-convolution. Before studying this property more thoroughly, we give it a formal name.

Definition 29: We say that a dominant permutation-symmetric probability-vector random variable $\bar{\mathbf{L}}$ is *balanced* if

$$\mathbb{E}\left[\bar{L}_g^2\right] = \mathbb{E}\left[\bar{L}_0\bar{L}_g\right]$$

for all $g \in \mathbb{F}_q$.

Proposition 30: Let $\bar{\mathbf{L}}$ be a dominant permutation-symmetric probability-vector random variable. Then, $\bar{\mathbf{L}}$ is *balanced* if and only if

$$\mathbb{E}\left[\left\|\bar{\mathbf{L}}\right\|_2^2\right] = \mathbb{E}\left[\bar{L}_0\right]. \quad (46)$$

This structure leads to a corollary that will become very important when computing the mean-squared-error (MSE) of graph messages.

Corollary 31: Let $\bar{\mathbf{L}}$ be a balanced dominant permutation-symmetric probability-vector random variable. Then,

$$\mathbb{E}\left[\|\bar{\mathbf{L}} - \mathbf{e}_0\|_2^2\right] = 1 - \mathbb{E}\left[\|\bar{\mathbf{L}}\|_2^2\right]. \quad (47)$$

Thus, when a graph message is a balanced dominant permutation-symmetric probability-vector random variable, the expected L_2-norm of that message is sufficient to compute the expected MSE of that same message. Understanding the close connection between the L_2 norm and the expected MSE, we now seek to compute the expected L_2 norm of the output of the $\mathbb{F}_q$ convolution operator.

Theorem 32: Suppose $\left\{\bar{\mathbf{L}}^{(p)}\right\}$ is a collection of independent, balanced dominant permutation-symmetric probability-vector random variables. For any natural number γ, the two-norm of $\bar{\mathbf{N}} = \bigodot_{p\in[\gamma]} \bar{\mathbf{L}}^{(p)}$ is given by

$$\mathbb{E}\left[\left\|\bar{\mathbf{N}}\right\|_2^2\right] = \mathbb{E}\left[\bar{N}_0\right] = \frac{1}{q} + \left(\frac{q}{q-1}\right)^{\gamma-1} \prod_{i\in[\gamma]} \left(\mathbb{E}\left[\left\|\bar{\mathbf{L}}^{(p)}\right\|_2^2\right] - \frac{1}{q}\right). \tag{48}$$

Furthermore, $\bar{\mathbf{N}}$ is itself a balanced dominant permutation-symmetric probability-vector random variable.

At this point, we pause to recall our goal of efficiently computing the expected MSE at the output of the denoiser given the MSE at its input. To accomplish this goal, we consider the notion of MSE message passing, whereby we construct a bipartite graph representing the constraints of the outer LDPC code, but whose edges all have unity weight. Then, we initialize each variable node with a local observation containing the expected starting MSE of that section. Then, we pass scalar expected MSE messages between variable nodes and check nodes, where each node approximates the expected output MSE given its input MSEs. After a fixed number of iterations, we then compute the output MSE, which is used in the computation of τ_{t+1}^2. In the next section, we use the results developed thusfar to define the message passing rules.

C. Computing the State Evolution

We begin with the first round of message passing in which variable to check node messages consist only of local observations. Under the all-zero codeword assumption, the messages arriving at any check node constitute a set of independent, balanced, dominant, permutation-symmetric Gaussian probability-vector random variables. Thus, Theorem 32 may be employed to obtain an exact expression for the expected two-norm of the resultant check to variable BP messages. This is consequential as it offers a means to calculate the MSE of a section estimate based on incoming BP messages.

Proposition 33: Let $\{\boldsymbol{\mu}_{v_j\to c_p} : v_j \in N(c_p)\setminus v_\ell\}$ constitute a set of independent, balanced, dominant, permutation-symmetric, Gaussian probability-vector random variables. Then, the MSE associated with leftbound message $\boldsymbol{\mu}_{c_p\to v_\ell}$ is equal to

$$\mathbb{E}\left[\left\|\mathbf{S}_\ell - \boldsymbol{\mu}_{c_p\to v_\ell}\right\|_2^2\right] = \frac{q-1}{q} - \left(\frac{q}{q-1}\right)^{\gamma-1} \prod_{\substack{v_j\in N(c_p)\\ v_j\neq v_\ell}} \left(\mathbb{E}\left[\left\|\boldsymbol{\mu}_{v_j\to c_p}\right\|_2^2\right] - \frac{1}{q}\right),$$

where $\gamma = |N(c_p)| - 1$.

We now turn our attention to the MSE of variable to check messages. The optimal BP message from variable node v_ℓ to check node c_p, defined in equation (26), is created as the Hadamard product of leftbound messages. Unfortunately, an exact characterization of the MSE associated with this operation remains elusive to the authors. It is challenging to calculate the MSE of this estimator due, in part, to the normalization step that appears in its construction. Furthermore, while it can be shown that $\boldsymbol{\mu}_{v_\ell\to c_p}$ is a dominant permutation symmetric probability-vector random variable if its inputs are similarly structured, it remains unclear whether this BP message is also balanced.

For the sake of tractability, we introduce a mild approximation for the incoming check to variable node messages. We begin by defining Ψ to be the bijection

$$\Psi\left(\tau_t^2\right) = \mathbb{E}\left[\boldsymbol{\alpha}_\ell(0)\right] = \mathbb{E}\left[\frac{e^{\frac{\mathbf{r}_\ell(0)}{\tau^2}}}{\sum_{h\in\mathbb{F}_q} e^{\frac{\mathbf{r}_\ell(h)}{\tau^2}}}\right], \tag{49}$$

where, under the all-zero codeword assumption, ℓ may be arbitrarily chosen from $[L]$. That is, Ψ uniquely associates each $\mathbb{E}[\boldsymbol{\alpha}(0)]$ with an effective noise variance τ_t^2. Using this bijection, we propose that each incoming check to variable node message $\boldsymbol{\mu}_{c_p\to v_\ell}$ be approximated as having been generated from a Gaussian model

$$\boldsymbol{\mu}_{c_p\to v_\ell}(g) \approx \tilde{\boldsymbol{\mu}}_{c_p\to v_\ell}(g) \sim \frac{e^{\frac{\mathbf{r}_{p,\ell}(g)}{\tau_{p,\ell}^2}}}{\sum_{h\in\mathbb{F}_q} e^{\frac{\mathbf{r}_{p,\ell}(h)}{\tau_{p,\ell}^2}}}. \tag{50}$$

Here, the p,ℓ notation indicates that the quantity of interest is associated with the approximate message from check node c_p to variable node v_ℓ. In (50), $\mathbf{r}_{p,\ell}$ is the hypothetical effective observation of section ℓ used to generate our approximation of $\boldsymbol{\mu}_{c_p\to v_\ell}$. Thus, $\mathbf{r}_{p,\ell} = \mathbf{s}_\ell + \mathbf{n}_{p,\ell}$, where $\mathbf{n}_{p,\ell}$ is zero-mean AWGN whose variance we will denote by $\tau_{p,\ell}^2$. The variance of $\mathbf{n}_{p,\ell}$ is obtained by finding the τ^2 value that would have generated $\boldsymbol{\mu}_{c_p\to v_\ell}$ had it truly been generated by the Gaussian model. Thus,

$$\tau_{p,\ell}^2 = \Psi^{-1}\left(\boldsymbol{\mu}_{c_p\to v_\ell}(0)\right). \tag{51}$$

Under this approximation, it becomes straightforward to track the MSE of the rightbound messages.

Proposition 34: Let $\{\tilde{\boldsymbol{\mu}}_{c_\xi\to v_\ell} : c_\xi \in N(v_\ell)\setminus c_p\}$ be a set of independent, balanced, dominant, permutation-symmetric, Gaussian probability-vector random variables having the distribution provided in (50). Then, the MSE associated with the rightbound message $\tilde{\boldsymbol{\mu}}_{v_\ell\to c_p}$ is given by

$$\mathbb{E}\left[\left\|\mathbf{S}_\ell - \tilde{\boldsymbol{\mu}}_{v_\ell\to c_p}\right\|_2^2\right] = 1 - \Psi\left(\tilde{\tau}_{\ell,p}^2\right). \tag{52}$$

Here, the effective noise variance $\tilde{\tau}_{\ell,p}^2$ is given by

$$\tilde{\tau}_{\ell,p}^2 = \frac{1}{\frac{1}{\tau^2} + \sum_{c_\xi\in N(v_\ell)\setminus c_p} \frac{1}{\tau_{\xi,\ell}^2}}, \tag{53}$$

where τ^2 denotes the effective noise variance of the effective observation from AMP and $\tau_{\xi,\ell}^2 = \Psi^{-1}\left(\mathbb{E}\left[\tilde{\boldsymbol{\mu}}_{c_\xi\to v_\ell}(0)\right]\right)$.

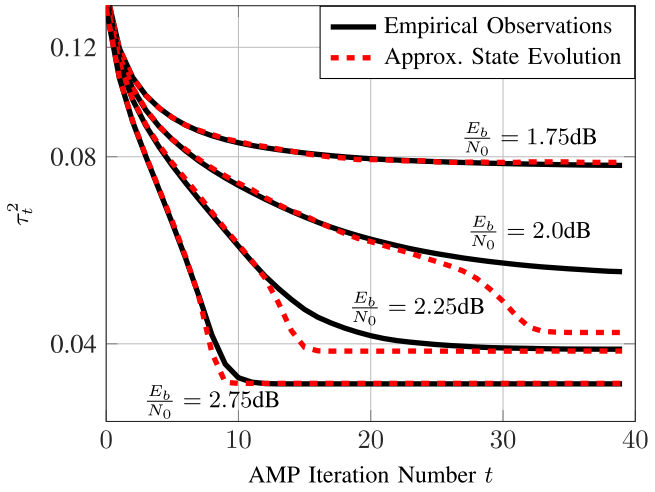

Fig. 4. Comparison between true τ_t^2 values and those predicted by the approximate state evolution algorithm. The approximate state evolution algorithm is very accurate for low and high SNRs but provides reduced insight when AMP is at the edge of its convergence region.

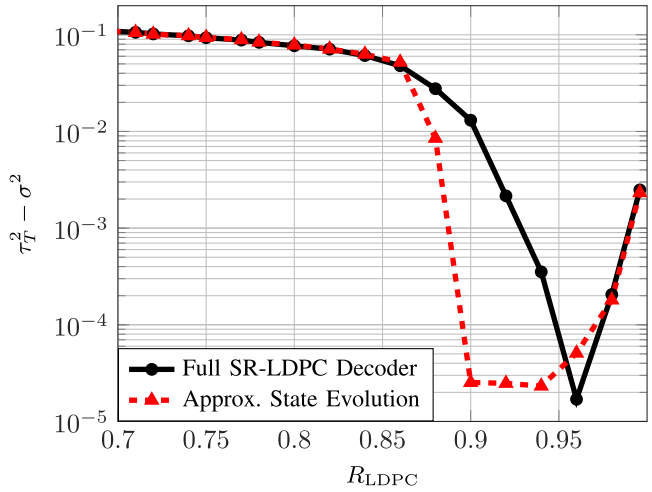

Fig. 5. Optimization of outer LDPC code rate under fixed overall rate constraint using the approximate state evolution algorithm. Approximate state evolution may be used as a coarse optimization technique, which may possibly be followed by fine tuning through a local rate search.

A similar result may be obtained for the output MSE of the denoiser by including all messages from neighboring check nodes as well as the local observation in the computation of $\tilde{\tau}_{\ell,p}^2$. Note that the approximate variable to check node message computed in Proposition 34 is also a balanced, dominant, permutation-symmetric, Gaussian probability-vector random variable. Furthermore, if Condition 9 is satisfied, the variable to check node messages received at a given check node will be independent. Thus, the MSE of the next round's check to variable messages can also be computed using Proposition 33, thus setting the stage for an iterative procedure.

Propositions 33 and 34 form the basis of the approximate state evolution algorithm of Definition 13. Note that each graph message in this MSE message-passing algorithm is a scalar; thus, this approximate state evolution algorithm is of a dimensionality that is orders of magnitude lower than the full SR-LDPC decoding algorithm. As will be shown in the next section, this property makes the approximate state evolution algorithm an attractive solution for hyperparameter tuning and code optimization tasks.

D. Performance of the State Evolution

Having defined a low-dimension approximate state evolution algorithm, we now seek to characterize its performance and identify its limitations. Before doing so, we note that a popular method for approximating the true value of τ_t^2 during AMP decoding is to use the following relation:

$$\tau_t^2 \approx \frac{\|\mathbf{z}^{(t)}\|_2^2}{n}. \tag{54}$$

As a benchmark, we thus run the full SR-LDPC decoder for a given code many times and average the τ_t^2 values computed from (54). Using this benchmark, we then compare the true τ_t^2 values with those predicted by the approximate state evolution algorithm for a specific SR-LDPC code. The details of this code will not be given in this section, but will be addressed in detail in Section V. Figure 4 highlights the results of this experiment.

From Fig. 4, we see that the $\hat{\tau}_t^2$ values predicted by the approximate state evolution algorithm are very accurate for low and high SNRs. When the $E_b/N_0 = 1.75$ dB, the SR-LDPC decoding algorithm fails to converge to the correct codeword; thus, $\tau_T^2 > \sigma^2$, or there is a non-zero fixed-point MSE. Clearly, the approximate state evolution algorithm identifies this fixed point MSE very well. Conversely, when the $E_b/N_0 = 2.75$ dB, the SR-LDPC decoding algorithm converges in the sense that $\tau_T^2 = \sigma^2$ and thus the final expected MSE is zero. However, when the SNR is such that the decoding algorithm is operating on the edge of its convergence region, the predictions provided by the approximate state evolution algorithm tend to be overconfident. In the example provided, at $E_b/N_0 = 2.0$ dB, the approximate state evolution algorithm predicts a τ_T^2 value that is lower than that observed in practice. Despite this caveat, the proposed approximate state evolution algorithm may be useful for code optimization as it enables the rapid comparison of different hyperparameter configurations (e.g. field size q, $\{N_t : t \in [T]\}$, outer LDPC code, etc) for a variety of SNRs. Of course, exact performance for the hyperparameter selected under approximate state evolution can be validated through full Monte-Carlo simulations of the SR-LDPC code.

As an example of using the approximate state evolution algorithm for hyperparameter tuning, consider the task of choosing the outer LDPC code rate R_{LDPC} while keeping the number of information bits and the number of channel uses fixed. In Fig. 5, we compare the performance of various SR-LDPC codes that are identical in every way except for their choices of outer codes. The SNR of this comparison is chosen to be $E_b/N_0 = 2.5$ dB, the number of AMP iterations is set at $T = 20$, and the difference $\hat{\tau}_T^2 - \sigma^2$ is plotted in Fig. 5 alongside the associated benchmark values. We choose to plot $\hat{\tau}_T^2 - \sigma^2$ because the σ^2 values are slightly different under each code and thus this difference provides a clearer comparison of the residual MSE. From Fig. 5, we see that the optimal $R_{\text{LDPC}} \approx 0.96$, which is relatively high. The intuition behind this phenomenon is that when R_{LDPC} is low, the undersampling

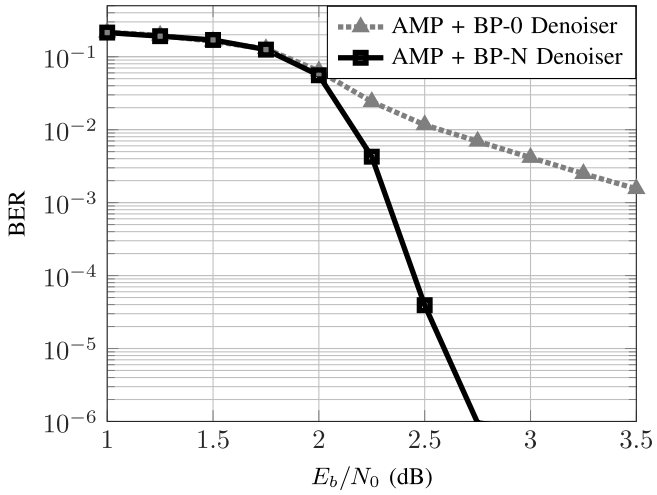

Fig. 6. BER of SR-LDPC code using AMP + BP-0 denoiser and AMP + BP-N denoiser. Incorporating the BP denoiser within AMP results in a steep waterfall in error performance.

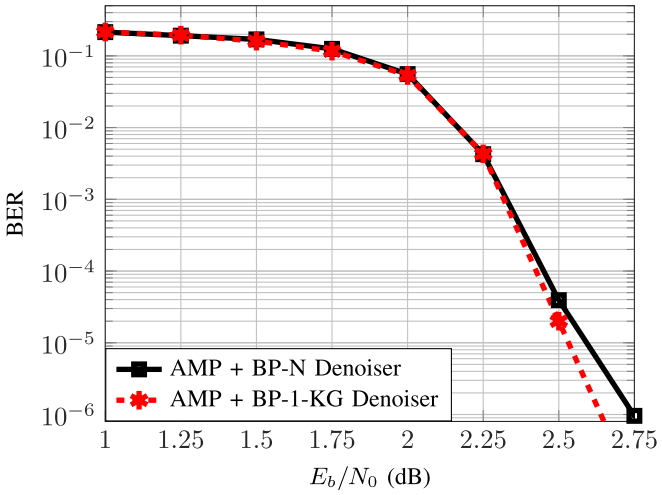

Fig. 7. BER of SR-LDPC code using AMP + BP-N and AMP + BP-1-KG denoisers. In this regime, the BP-1-KG denoiser reduces complexity without hurting performance.

ratio of AMP, or the ratio of the number of measurements to the dimensionality of the sparse vector $\delta = \frac{n}{qL}$, is so small that AMP may not be operating in its convergence region [34]. Conversely, when R_{LDPC} is high, the LDPC code has minimal error-correcting capabilities. The rate $R_{\text{LDPC}} \approx 0.96$ thus appears to offer the best tradeoff between competing design criteria. As before, we see that the approximate state evolution algorithm tends to be overconfident when operating on the edge of AMP's convergence region. Nevertheless, it is clear that the approximate state evolution algorithm significantly narrows down the search space for the optimal R_{LDPC} and thus can be used as a coarse optimization tool, where more precise optimization may be done via full Monte-Carlo (MC) simulations.

V. Simulation Results

In this section, we investigate the performance of SR-LDPC codes.[1] Specifically, we simulate a randomly generated SR-LDPC code that encodes 5888 information bits into 7350 coded symbols. We define the rate of the SR-LDPC code to be the number of information bits over the number of channel uses; thus, we have that $R_{\text{SRLDPC}} \approx 0.80$. The non-binary LDPC code employed is a $(766, 736)$ code of rate $R_{\text{LDPC}} \approx 0.96$ over GF(256) whose edges are generated via progressive edge growth (PEG) and whose weights are chosen uniformly at random from the elements of $\mathbb{F}_{256} \setminus 0$. These parameters were chosen to facilitate comparisons with similar codes in the literature. Throughout this section, we run 25 AMP iterations using the BP denoiser followed by 100 final BP iterations. If at any point during the decoding process a valid codeword is obtained, the decoding process is terminated.

Recall that the BP denoiser is parameterized by $\{N_t : t \in [T]\}$, or the schedule of the number of BP iterations to perform per AMP iteration. In this section, we will investigate various schedules for the BP denoiser. To begin our study, we consider the BP-N denoiser, which is defined as the BP denoiser in which $N_t = t+1$ for every AMP iteration $t = 0, 1, 2, \ldots T-1$.

[1]The source code used to generate these results is available online at https://github.com/EngProjects/mMTC/tree/code

Before comparing the performance of our SR-LDPC code to other error-correcting codes, we first seek to evaluate the performance of AMP with the dynamic denoiser from Definition 8. To do this, we introduce a second denoiser, which we refer to as the BP-0 denoiser, as this denoiser performs zero rounds of BP per AMP iteration, or $N_t = 0 \ \forall t$. Note that, under this denoiser, no information is shared between inner and outer decoders. We then compute and compare the BER performance of our randomly-generated SR-LDPC code under AMP + BP-N decoding and AMP + BP-0 decoding. In both cases, we run 100 rounds of BP after the AMP-BP process has terminated. In essence, this experiment compares the performance of jointly decoding the inner and outer codes vs decoding the inner, then outer codes in disjoint succession. Fig. 6 compares the bit error rate (BER) performance of the SR-LDPC code with the BP-N denosier and the BP-0 denoiser. Clearly, AMP + BP-N decoding endows the SR-LDPC code with a steep waterfall in BER, a phenomenon not seen in AMP + BP-0 decoding. We thus conclude that the proposed joint decoder is superior to a disjoint decoder in terms of error performance.

The intuition behind the BP-N denoiser becomes clear when one considers how the effective noise variance τ_t^2 of $\mathbf{r}^{(t)}$ is ideally decreasing with AMP iteration count t, up to a certain point. During the first few AMP iterations, the outer factor graph is initialized with very noisy local observations so consequently, BP cannot improve $\mathbf{r}^{(t)}$ very much, even if many BP iterations are run. However, as t increases, τ_t^2 hopefully decreases, so BP is able to meaningfully improve the quality of the state estimates. Ideally, τ_t^2 will eventually fall below the BP threshold of the outer LDPC code, at which point BP should be run until BP decoding succeeds. In practice, conditions are not always ideal; nevertheless, increasing N_t with t seems to make sense. Though intuitive, we make no claims that the BP-N strategy is optimal, and we leave the optimal scheduling of BP iterations as an open problem for future work. Though we decide N_t based solely on t, we note that the optimal N_t^* may also depend on τ_t^2.

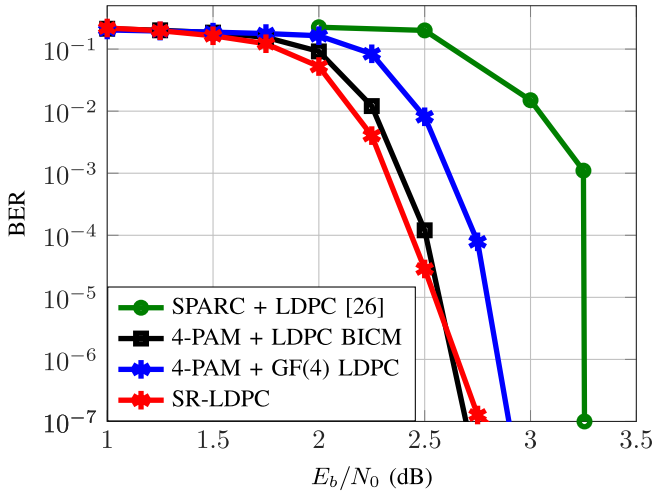

Fig. 8. BER comparison between an SR-LDPC code, the SPARC + LDPC concatenated code from [26], a BICM scheme involving 4-PAM and NR LDPC codes, and a 4-PAM + GF(4) LDPC scheme.

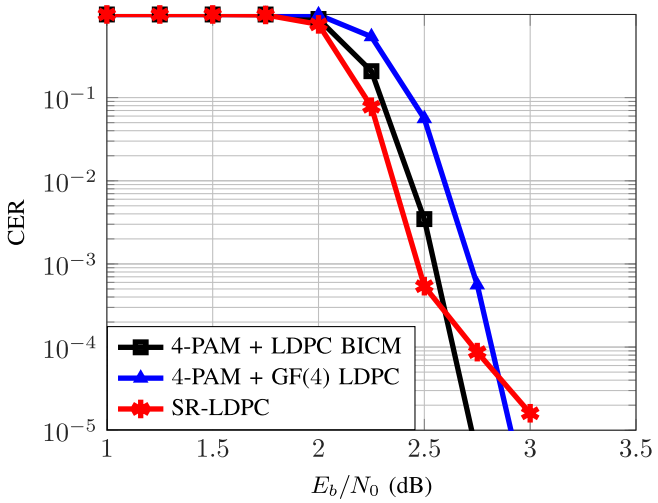

Fig. 9. CER performance comparison between an SR-LDPC code, a BICM scheme involving 4-PAM and NR LDPC codes, and a 4-PAM + GF(4) LDPC scheme.

Furthermore, though the BP-N strategy is elegant, it requires $T(T+1)/2 = \mathcal{O}\left(T^2\right)$ total BP iterations, where T is the number of AMP iterations to perform. For even moderate T, the complexity of this decoding strategy becomes significant. As an alternative approach, we propose the BP-1-KeepGraph (BP-1-KG) denoiser in which only one BP iteration is run per AMP iteration ($N_t = 1\ \forall t$). However, instead of completely resetting all graph messages and local observations between AMP iterations, only the local observations are reset and the graph messages from the previous round are left to be incorporated into the current round's message passing. Thus, this strategy requires only $\mathcal{O}(T)$ BP iterations. Essentially, this approach uses noisy messages from previous rounds as information from further down the computation tree instead of passing fresh information across the entire computation tree during every AMP iteration. Figure 7 compares the performance of the SR-LDPC code using the BP-N an d BP-1-KG denoisers. Demonstrably, there is minimal degradation in BER performance when the BP-1-KG denoiser is used at low SNRs and, somewhat surprisingly, a performance boost at high SNRs; thus, the BP-1-KG denoiser may be used as a pragmatic means to reduce decoding complexity.

We now seek to compare the performance of the SR-LDPC code using the BP-1-KG denoiser to three pertinent benchmarks: a highly-optimized SPARC/LDPC construction from [26], a bit-interleaved coded modulation (BICM) scheme involving Gray-coded 4-PAM and a rate $R = 0.4$ NR LDPC code, and Gray-coded 4-PAM with a GF(4) non-binary LDPC code generated using an irregular degree distribution from [47]. Before proceeding, we pause to emphasize that SR-LDPC codes may provide Gaussian inputs to the AWGN channel, while the latter two benchmarks provide equiprobable 4-PAM inputs. In general, the capacity of the AWGN channel with Gaussian inputs is higher than the capacity of the AWGN channel with $4-$PAM inputs; however, at our chosen rate of $R = 0.8$, the gap between these two capacities is negligible [48]. Thus, the observed performance differences are not caused by fundamental limitations of the underlying AWGN channel. A BER comparison of these three schemes is included in Fig. 8 and a corresponding CER comparison is included in Fig. 9. Note that the CER of the scheme from [26] is not included in Fig. 9 as it was not provided in Greig and Venkataramanan's original paper.

From these figures, it is clear that this SR-LDPC code provides an improvement of about 1 dB at a BER of 10^{-3} over the SPARC/LDPC concatenated coding structure from [26]. Furthermore, the SR-LDPC code outperforms the 4-PAM with GF(4) LDPC scheme by just under 0.5 dB and the BICM scheme by about 0.1 dB at that same BER. In terms of CER, the SR-LDPC code provides about a 0.3 dB improvement over the 4-PAM with GF(4) LDPC scheme and an improvement of under 0.1 dB over the 4-PAM + LDPC with BICM at a CER of 10^{-1}, which is a common target CER when an ARQ outer loop is employed. We note that the performance gap between the BICM and non-binary LDPC may be reduced through further optimization of the non-binary LDPC code. Interestingly, this SR-LDPC code was not thoroughly optimized while the code in [26] and the NR LDPC codes are both highly optimized. Thus, it is likely that further performance improvements are possible through the careful design of the SR-LDPC code.

Though no error floor is observed in Figures 8 and 9, such a floor may exist at finite blocklengths. However, it appears that the SR-LDPC decoding process is propitious to having a relatively low error floor. Specifically, the outer non-binary LDPC code serves to enforce global parity consistency throughout the codeword, which steepens the waterfall region and lowers the error floor when compared to uncoded SPARCs. Furthermore, though a high-rate outer non-binary LDPC code may in general be at a greater risk of trapping sets, each AMP iteration in SR-LDPC generates a fresh effective observation $\mathbf{r}^{(t)}$ and therefore gives the BP denoiser a chance to improve the likelihoods. This is in stark contrast with the regular BP process. Preliminary evidence points to performance improvement for LDPC decoding with fresh observations. Still, the outer LDPC code is at risk of low-weight codewords, which increase the probability of converging to a valid, but incorrect, codeword.

As SR-LDPC decoding terminates when a valid codeword is obtained, this phenomenon does contribute to errors.

Finally, recall that the entries of the sensing matrix $\mathbf{A}$ are generated as i.i.d. $\mathcal{N}\left(0, \frac{1}{n}\right)$ random variables. As each SR-LDPC codeword is a linear combination of the columns of $\mathbf{A}$, every channel input is therefore the realization of a $\mathcal{N}\left(0, \frac{L}{n}\right)$ Gaussian random variable. It is well-known that the capacity of the AWGN channel is achieved with a Gaussian input distribution, and that the performance of coded modulation schemes may be improved by shaping the constellation to be Gaussian-like [48]. While much work has been done on forcing traditional constellations (e.g., M-QAM) to be Gaussian-like, SR-LDPC coding is a natural strategy to combine Gaussian signalling with traditional codes in a powerful way. Thus, we view SR-LDPC coding as a pragmatic strategy for obtaining shaping gains over the AWGN channel.

VI. Conclusion

This article introduces sparse regression LDPC (SR-LDPC) codes and their decoding. SR-LDPC codes are formed by concatenating an inner sparse regression code with an outer non-binary LDPC code whose respective field size and section sizes are equal. Such codes can be efficiently decoded using AMP with a dynamic denoiser that runs BP on the factor graph of the outer LDPC code, thus allowing for soft information to be shared between inner and outer decoders. It is shown that the proposed BP denoiser falls within the framework of non-separable denoising functions and subsequently, that state evolution holds for the proposed AMP-BP algorithm. Furthermore, by exploiting the structure of SR-LDPC codes and the proposed decoding algorithm, a computationally efficient approximate state evolution recursion is presented that allows for rapid code optimization and hyperparameter tuning.

Numerical simulation results are presented to demonstrate that the proposed AMP-BP decoder, which jointly decodes inner and outer codes, significantly outperforms a traditional Forney-style decoding algorithm. Additionally, an SR-LDPC code is shown to outperform contemporary codes over the AWGN channel.

The results presented in this article were obtained for a relatively unoptimized SR-LDPC code. Thus, it is likely that further performance improvements may be obtained through proper optimization of the code structure. For example, it is known that the performance of uncoded SPARCs can be significantly improved via a nonuniform power allocation; yet, in this article, we employ a naive uniform power allocation. Thus, the optimal power allocation for SR-LDPC codes remains a promising open problem. Other open problems include the optimal design of the outer LDPC code and the optimal number of BP iterations to perform per AMP iteration.

Appendix A
Properties of BP Denoiser

In this appendix, we prove that the BP-N denoiser is uniformly Lipschitz continuous under certain assumptions (Theorem 11) and we derive the Onsager correction term associated with the BP denoiser (Proposition 12).

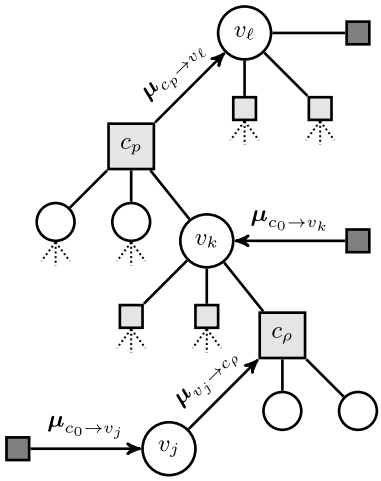

Fig. 10. Computation tree for variable node v_ℓ obtained by taking the factor graph of the outer LDPC code, setting v_ℓ as the root, and retaining only the nodes involved in the computation of $\hat{\mathbf{s}}_\ell(\mathbf{r}, g)$. We use this data structure to compute the derivatives in (56).

A. Proof of Theorem 11

One of the conditions for state evolution to hold for non-separable functions is that the denoiser must be uniformly-Lipschitz with respect to its ambient dimension [35]. The main objective of this section is to demonstrate that this property holds for the proposed BP denoiser under Conditions 9 and 10. To achieve this goal, our strategy is to demonstrate that the magnitudes of the entries in the Jacobian matrix of $\boldsymbol{\eta}(\mathbf{r})$ with respect to $\mathbf{r}$ are uniformly bounded.

Recall that the denoiser assumes a sectional form, as described in Definition 8. The vector estimate for section ℓ is

$$\hat{\mathbf{s}}_\ell(\mathbf{r}) = \sum_{g \in \mathbb{F}_q} \Pr\left(V_\ell = g | \mathbf{R}_{\text{tree}} = \mathbf{r}_{\text{tree}}\right) \mathbf{e}_g,$$

where $\mathbf{R}_{\text{tree}}$ denotes the measurements associated with the computational tree of the LDPC code rooted at section ℓ. The (realized) scaling factors found in (27) are given by

$$\hat{\mathbf{s}}_\ell(\mathbf{r}) = \frac{\bigcirc_{c_p \in N_0(v_\ell)} \boldsymbol{\mu}_{c_p \to v_\ell}}{\left\| \bigcirc_{c_p \in N_0(v_\ell)} \boldsymbol{\mu}_{c_p \to v_\ell} \right\|_1}. \tag{55}$$

where $N_0(v_\ell)$ denotes the neighborhood of v_ℓ including the local observation and $\boldsymbol{\mu}_{c_0 \to v_\ell} = \boldsymbol{\alpha}_\ell$. We are ultimately interested in Jacobian entries of the form

$$\begin{aligned} \frac{\partial \hat{\mathbf{s}}_\ell(\mathbf{r}, g)}{\partial \mathbf{r}_j(h)} &= \frac{\partial}{\partial \mathbf{r}_j(h)} \frac{\prod_{c_p \in N_0(v_\ell)} \boldsymbol{\mu}_{c_p \to v_\ell}(g)}{\left\| \bigcirc_{c_p \in N_0(v_\ell)} \boldsymbol{\mu}_{c_p \to v_\ell} \right\|_1} \\ &= \frac{\partial}{\partial \mathbf{r}_j(h)} \frac{\boldsymbol{\alpha}_\ell(g) \prod_{c_p \in N(v_\ell)} \boldsymbol{\mu}_{c_p \to v_\ell}(g)}{\sum_{k \in \mathbb{F}_q} \boldsymbol{\alpha}_\ell(k) \prod_{c_p \in N(v_\ell)} \boldsymbol{\mu}_{c_p \to v_\ell}(k)} \end{aligned} \tag{56}$$

for $g, h \in \mathbb{F}_q$ and $\ell, j \in [L]$. We adopt a divide-and-conquer approach to identify and bound these derivatives. Specifically, we focus on the rooted tree obtained by taking the factor graph of the outer LDPC code, setting v_ℓ as the root, and retaining only the nodes involved in the computation of $\hat{\mathbf{s}}_\ell(\mathbf{r}, g)$. Under Condition 9, this sub-graph must form a proper tree; Fig. 10 offers a notional diagram of this concept.

We seek to bound the magnitude of the derivatives in (56) based on the distance between v_ℓ and v_j in this rooted tree. We begin with local observations.

Proposition 35 (Local Observations): The partial derivatives of $\boldsymbol{\alpha}_j$ with respect to $\mathbf{r}_k(h)$ are given by

$$\frac{\partial \boldsymbol{\alpha}_j}{\partial \mathbf{r}_k(h)} = \begin{cases} \frac{1}{\tau^2}\boldsymbol{\alpha}_j(h)\left(\mathbf{e}_h - \boldsymbol{\alpha}_j\right) & j = k \\ 0 & j \neq k \end{cases} \tag{57}$$

for $h \in \mathbb{F}_q$ and where $\tau > 0$ is the standard deviation of the effective observation.

Proof: As defined in (14), the vector $\boldsymbol{\alpha}_j$ is given by

$$\boldsymbol{\alpha}_j(g) = \frac{e^{\frac{\mathbf{r}_j(g)}{\tau^2}}}{\sum_{k\in\mathbb{F}_q} e^{\frac{\mathbf{r}_j(k)}{\tau^2}}} \qquad \forall g \in \mathbb{F}_q.$$

When $j \neq k$, it immediately follows that $\partial\boldsymbol{\alpha}_j/\partial\mathbf{r}_k(h) = 0$ as $\boldsymbol{\alpha}_j$ does not depend on $\mathbf{r}_k(h)$. We thus consider the case where $k = j$. When $g = h$, we have that

$$\begin{aligned}\frac{\partial \boldsymbol{\alpha}_j(h)}{\partial \mathbf{r}_j(h)} &= \frac{1}{\tau^2}\frac{e^{\frac{\mathbf{r}_j(h)}{\tau^2}}}{\sum_{k\in\mathbb{F}_q} e^{\frac{\mathbf{r}_j(k)}{\tau^2}}} - \frac{1}{\tau^2}\frac{e^{\frac{\mathbf{r}_j(h)}{\tau^2}} e^{\frac{\mathbf{r}_j(h)}{\tau^2}}}{\left(\sum_{k\in\mathbb{F}_q} e^{\frac{\mathbf{r}_j(k)}{\tau^2}}\right)^2} \\ &= \frac{1}{\tau^2}\boldsymbol{\alpha}_j(h)\left(1 - \boldsymbol{\alpha}_j(h)\right).\end{aligned}$$

When $g \neq h$, we get

$$\frac{\partial \boldsymbol{\alpha}_j(g)}{\partial \mathbf{r}_j(h)} = -\frac{1}{\tau^2}\frac{e^{\frac{\mathbf{r}_j(g)}{\tau^2}} e^{\frac{\mathbf{r}_j(h)}{\tau^2}}}{\left(\sum_{\kappa\in\mathbb{F}_q} e^{\frac{\mathbf{r}_j(\kappa)}{\tau^2}}\right)^2} = -\frac{1}{\tau^2}\boldsymbol{\alpha}_j(g)\boldsymbol{\alpha}_j(h).$$

Collecting these findings and condensing them into a more compact form, we arrive at (57), which is the desired expression. ∎

Corollary 36: The absolute value of the partial derivatives of $\boldsymbol{\alpha}_j$ with respect to $\mathbf{r}_k(h)$ are bounded by

$$\left|\frac{\partial \boldsymbol{\alpha}_j}{\partial \mathbf{r}_k(h)}\right| \leq \frac{1}{4\tau^2} \tag{58}$$

where $\tau > 0$ is the standard deviation of the effective observation.

The proof of this corollary is trivial when $\boldsymbol{\alpha}_j$ is a valid probability vector, as is the case in this article. We also note that, based on the state evolution of AMP, $\tau^2 \geq \sigma^2$ at every iteration irrespective of the iteration number. We can therefore establish a uniform bound across iterations. We are now ready to show that the absolute value of (56) is bounded whenever $\ell = j$, i.e., at the root level of the computation tree.

Proposition 37 (Root Derivatives): The partial derivatives of $\hat{\mathbf{s}}_\ell(\mathbf{r}, g)$ with respect to $\mathbf{r}_\ell(h)$ are given by

$$\frac{\partial \hat{\mathbf{s}}_\ell(\mathbf{r})}{\partial \mathbf{r}_\ell(h)} = \frac{1}{\tau^2}\hat{\mathbf{s}}_\ell(\mathbf{r}, h)\left(\mathbf{e}_h - \hat{\mathbf{s}}_\ell(\mathbf{r})\right) \qquad \forall h \in \mathbb{F}_q \tag{59}$$

where $\tau > 0$ is the standard deviation of the effective observation.

Proof: Leveraging Proposition 35 and denoting the standard inner product by $\langle\cdot,\cdot\rangle$, we have

$$\begin{aligned}\frac{\partial \hat{\mathbf{s}}_\ell(\mathbf{r})}{\partial \mathbf{r}_\ell(h)} &= \frac{\partial}{\partial \mathbf{r}_\ell(h)}\frac{\boldsymbol{\alpha}_\ell \circ \bigcirc_{c_p\in N(v_\ell)} \boldsymbol{\mu}_{c_p\to v_\ell}}{\left\|\boldsymbol{\alpha}_\ell \circ \bigcirc_{c_p\in N(v_\ell)} \boldsymbol{\mu}_{c_p\to v_\ell}\right\|_1} \\ &= \frac{\frac{\partial \boldsymbol{\alpha}_\ell}{\partial \mathbf{r}_\ell(h)} \circ \left(\bigcirc_{c_p\in \mathrm{N}(v_\ell)} \boldsymbol{\mu}_{c_p\to v_\ell}\right)}{\left\|\boldsymbol{\alpha}_\ell \circ \left(\bigcirc_{c_p\in \mathrm{N}(v_\ell)} \boldsymbol{\mu}_{c_p\to v_\ell}\right)\right\|_1} \\ &\quad - \hat{\mathbf{s}}_\ell(\mathbf{r})\frac{\left\langle \frac{\partial \boldsymbol{\alpha}_\ell}{\partial \mathbf{r}_\ell(h)}, \bigcirc_{c_p\in \mathrm{N}(v_\ell)} \boldsymbol{\mu}_{c_p\to v_\ell}\right\rangle}{\left\|\boldsymbol{\alpha}_\ell \circ \left(\bigcirc_{c_p\in \mathrm{N}(v_\ell)} \boldsymbol{\mu}_{c_p\to v_\ell}\right)\right\|_1} \\ &= \frac{\boldsymbol{\alpha}_\ell(h)}{\tau^2}\frac{\left(\mathbf{e}_h - \boldsymbol{\alpha}_\ell\right) \circ \left(\bigcirc_{c_p\in \mathrm{N}(v_\ell)} \boldsymbol{\mu}_{c_p\to v_\ell}\right)}{\left\|\boldsymbol{\alpha}_\ell \circ \left(\bigcirc_{c_p\in \mathrm{N}(v_\ell)} \boldsymbol{\mu}_{c_p\to v_\ell}\right)\right\|_1} \\ &\quad - \frac{\boldsymbol{\alpha}_\ell(h)}{\tau^2}\hat{\mathbf{s}}_\ell(\mathbf{r})\frac{\left\langle \mathbf{e}_h - \boldsymbol{\alpha}_\ell, \bigcirc_{c_p\in \mathrm{N}(v_\ell)} \boldsymbol{\mu}_{c_p\to v_\ell}\right\rangle}{\left\|\boldsymbol{\alpha}_\ell \circ \left(\bigcirc_{c_p\in \mathrm{N}(v_\ell)} \boldsymbol{\mu}_{c_p\to v_\ell}\right)\right\|_1} \\ &= \frac{\boldsymbol{\alpha}_\ell(h)}{\tau^2}\frac{\mathbf{e}_h \circ \left(\bigcirc_{c_p\in \mathrm{N}(v_\ell)} \boldsymbol{\mu}_{c_p\to v_\ell}\right)}{\left\|\boldsymbol{\alpha}_\ell \circ \left(\bigcirc_{c_p\in \mathrm{N}(v_\ell)} \boldsymbol{\mu}_{c_p\to v_\ell}\right)\right\|_1} \\ &\quad - \frac{\boldsymbol{\alpha}_\ell(h)}{\tau^2}\hat{\mathbf{s}}_\ell(\mathbf{r})\frac{\left\langle \mathbf{e}_h, \bigcirc_{c_p\in \mathrm{N}(v_\ell)} \boldsymbol{\mu}_{c_p\to v_\ell}\right\rangle}{\left\|\boldsymbol{\alpha}_\ell \circ \left(\bigcirc_{c_p\in \mathrm{N}(v_\ell)} \boldsymbol{\mu}_{c_p\to v_\ell}\right)\right\|_1} \\ &= \frac{1}{\tau^2}\hat{\mathbf{s}}_\ell(\mathbf{r}, h)\left(\mathbf{e}_h - \hat{\mathbf{s}}_\ell(\mathbf{r})\right),\end{aligned}$$

which is the desired expression. ∎

Corollary 38: The absolute value of the partial derivatives of $\hat{\mathbf{s}}_\ell(\mathbf{r})$ with respect to $\mathbf{r}_\ell(h)$ are bounded by

$$\left|\frac{\partial \hat{\mathbf{s}}_\ell(\mathbf{r})}{\partial \mathbf{r}_\ell(h)}\right| \leq \frac{1}{4\tau^2} \tag{60}$$

The proof of this corollary follows that of Corollary 36 because, like $\boldsymbol{\alpha}_j$, $\hat{\mathbf{s}}_\ell(\mathbf{r})$ forms a valid probability vector.

Proposition 37 offers a blueprint for the general result we wish to establish. Yet, the situation gets more complicated when $\ell \neq j$ because we have to involve the message passing rules. In doing so, we obtain a key intermediate result using mathematical induction. We start with the variable node closest to the root node, and then progress outward step by step.

To circumvent a notational nightmare, we restrict the proof to cases where all edge weights are equal to $1 \in \mathbb{F}_q$. Conceptually, the edge can be interpreted as permutations on the belief vectors. From the point of view of bounding partial derivatives, this is a benign operation, yet the accounting that comes with permutations is dreadful, hence our focus on the simpler case. Moving forward, we assume the following condition.

Condition 39: All edge weights within the factor graph of the LDPC outer code are equal to $1 \in \mathbb{F}_q$.

The extension of the following propositions to the case with arbitrary edge weights (i.e., beyond Condition 39) is conceptually straightforward.

Proposition 40: Suppose Condition 9 holds and let v_j be a descendant of root node v_ℓ in the computation tree. Moreover, let $c_p \in N(v_\ell)$ be the unique check neighbor of v_ℓ on the path from v_ℓ to v_j within the tree. Then, there exists vector $\boldsymbol{\nu}$, with $\mathbf{0} \preceq \boldsymbol{\nu} \preceq \boldsymbol{\mu}_{c_p\to v_\ell}$, such that the partial derivative of $\boldsymbol{\mu}_{c_p\to v_\ell}$ with respect to $\mathbf{r}_j(h)$ is given by

$$\frac{\partial \boldsymbol{\mu}_{c_p\to v_\ell}}{\partial \mathbf{r}_j(h)} = \frac{1}{\tau^2}\left(\boldsymbol{\nu} - \|\boldsymbol{\nu}\|_1 \boldsymbol{\mu}_{c_p\to v_\ell}\right), \tag{61}$$

where $\tau > 0$ is the standard deviation of the effective observation. Here, $\preceq$ denotes elementwise comparison of the entries in the vector.

Proof: Under Condition 9, we know that v_j appears at most once within the computation tree rooted at v_ℓ. Thus, we establish (61) via mathematical induction on the distance between v_ℓ and its descendant v_j on the computation tree. The distance that we are interested in only considers the number of variable nodes between v_ℓ and v_j. Before beginning, we point out that if v_j is not a descendant of v_ℓ, then the corresponding partial derivatives vanish and the claim is immediate.

We begin with generic results that are useful for both the base case and the inductive step. Let v_j be a descendant of v_ℓ and let v_k be the variable child of v_ℓ on the path from v_ℓ to v_j. Let c_p be the unique check node in $N(v_\ell) \cap N(v_k)$ and let c_ϱ be the unique check node child of v_k on the path from v_k to v_j; if $v_k = v_j$, let $\varrho = 0$. Then, we have that

$$\frac{\partial \boldsymbol{\mu}_{v_k \to c_p}}{\partial \mathbf{r}_j(h)} = \frac{\partial}{\partial \mathbf{r}_j(h)} \frac{\bigcirc_{c_\xi \in N_0(v_k) \setminus c_p} \boldsymbol{\mu}_{c_\xi \to v_k}}{\left\| \bigcirc_{c_\xi \in N_0(v_k) \setminus c_p} \boldsymbol{\mu}_{c_\xi \to v_k} \right\|_1}. \tag{62}$$

We can also examine the partial derivatives of the probability vector $\boldsymbol{\mu}_{c_\varrho \to v_k}$. Suppose $v_k \neq v_j$ and let v_o be the unique variable child of v_k on the path from v_ℓ to v_j. Using the $\mathbb{F}_q$ convolution, we have

$$\begin{aligned}
\frac{\partial \boldsymbol{\mu}_{c_\varrho \to v_k}}{\partial \mathbf{r}_j(h)} &= \frac{\partial}{\partial \mathbf{r}_j(h)} \left(\bigodot_{v_l \in N(c_\varrho) \setminus v_k} \boldsymbol{\mu}_{v_l \to c_\varrho} \right) \\
&= \frac{\partial \boldsymbol{\mu}_{v_o \to c_\varrho}}{\partial \mathbf{r}_j(h)} \odot \left(\bigodot_{v_l \in N(c_\varrho) \setminus \{v_k, v_o\}} \boldsymbol{\mu}_{v_l \to c_\varrho} \right) \\
&= \frac{\partial \boldsymbol{\mu}_{v_o \to c_\varrho}}{\partial \mathbf{r}_j(h)} \odot \boldsymbol{\nu}_{c_\varrho \setminus v_k, v_o}.
\end{aligned} \tag{63}$$

We emphasize that $\boldsymbol{\nu}_{c_\varrho \setminus v_k, v_o}$, as defined implicitly above, is a probability distribution.

Having established these results, we turn our attention to the base case where v_j is a variable child of v_ℓ ($v_k = v_j$, $\varrho = 0$). Applying (62) and Proposition 35, we obtain

$$\begin{aligned}
\frac{\partial \boldsymbol{\mu}_{v_j \to c_p}}{\partial \mathbf{r}_j(h)} &= \frac{\partial}{\partial \mathbf{r}_j(h)} \frac{\boldsymbol{\alpha}_j \circ \left(\bigcirc_{c_\xi \in N(v_j) \setminus c_p} \boldsymbol{\mu}_{c_\xi \to v_j} \right)}{\left\| \boldsymbol{\alpha}_j \circ \left(\bigcirc_{c_\xi \in N(v_j) \setminus c_p} \boldsymbol{\mu}_{c_\xi \to v_j} \right) \right\|_1} \\
&= \frac{\frac{\partial \boldsymbol{\alpha}_j}{\partial \mathbf{r}_j(h)} \circ \left(\bigcirc_{c_\xi \in N(v_j) \setminus c_p} \boldsymbol{\mu}_{c_\xi \to v_j} \right)}{\left\langle \boldsymbol{\alpha}_j, \bigcirc_{c_\xi \in N(v_j) \setminus c_p} \boldsymbol{\mu}_{c_\xi \to v_j} \right\rangle} \\
&\quad - \boldsymbol{\mu}_{v_j \to c_p} \frac{\left\langle \frac{\partial \boldsymbol{\alpha}_j}{\partial \mathbf{r}_j(h)}, \bigcirc_{c_\xi \in N(v_j) \setminus c_p} \boldsymbol{\mu}_{c_\xi \to v_j} \right\rangle}{\left\langle \boldsymbol{\alpha}_j, \bigcirc_{c_\xi \in N(v_j) \setminus c_p} \boldsymbol{\mu}_{c_\xi \to v_j} \right\rangle} \\
&= \frac{1}{\tau^2} \frac{\boldsymbol{\alpha}_j(h) \mathbf{e}_h \circ \left(\bigcirc_{c_\xi \in N(v_j) \setminus c_p} \boldsymbol{\mu}_{c_\xi \to v_j} \right)}{\left\langle \boldsymbol{\alpha}_j, \bigcirc_{c_\xi \in N(v_j) \setminus c_p} \boldsymbol{\mu}_{c_\xi \to v_j} \right\rangle} \\
&\quad - \frac{1}{\tau^2} \boldsymbol{\mu}_{v_j \to c_p} \frac{\left\langle \boldsymbol{\alpha}_j(h) \mathbf{e}_h, \bigcirc_{c_\xi \in N(v_j) \setminus c_p} \boldsymbol{\mu}_{c_\xi \to v_j} \right\rangle}{\left\langle \boldsymbol{\alpha}_j, \bigcirc_{c_\xi \in N(v_j) \setminus c_p} \boldsymbol{\mu}_{c_\xi \to v_j} \right\rangle} \\
&= \frac{1}{\tau^2} \left(\boldsymbol{\nu}_{v_j \to c_p} - \left\| \boldsymbol{\nu}_{v_j \to c_p} \right\|_1 \boldsymbol{\mu}_{v_j \to c_p} \right),
\end{aligned} \tag{64}$$

where

$$\boldsymbol{\nu}_{v_j \to c_p} = \frac{\boldsymbol{\alpha}_j(h) \mathbf{e}_h \circ \left(\bigcirc_{c_\xi \in N(v_j) \setminus c_p} \boldsymbol{\mu}_{c_\xi \to v_j} \right)}{\left\langle \boldsymbol{\alpha}_j, \bigcirc_{c_\xi \in N(v_j) \setminus c_p} \boldsymbol{\mu}_{c_\xi \to v_j} \right\rangle}.$$

By construction, we have $\mathbf{0} \preceq \boldsymbol{\nu}_{v_j \to c_p} \preceq \boldsymbol{\mu}_{v_j \to c_p}$. We turn to the second graph operation and apply (63), which yields

$$\begin{aligned}
\frac{\partial \boldsymbol{\mu}_{c_p \to v_\ell}}{\partial \mathbf{r}_j(h)} &= \frac{\partial \boldsymbol{\mu}_{v_j \to c_p}}{\partial \mathbf{r}_j(h)} \odot \boldsymbol{\nu}_{c_p \setminus v_\ell, v_j} \\
&= \frac{1}{\tau^2} \left(\boldsymbol{\nu}_{v_j \to c_p} - \left\| \boldsymbol{\nu}_{v_j \to c_p} \right\|_1 \boldsymbol{\mu}_{v_j \to c_p} \right) \odot \boldsymbol{\nu}_{c_p \setminus v_\ell, v_j} \\
&= \frac{1}{\tau^2} \left(\boldsymbol{\nu}_{v_j \to c_p} \odot \boldsymbol{\nu}_{c_p \setminus v_\ell, v_j} - \left\| \boldsymbol{\nu}_{v_j \to c_p} \right\|_1 \boldsymbol{\mu}_{c_p \to v_\ell} \right).
\end{aligned} \tag{65}$$

Thus, in this case, we take $\boldsymbol{\nu} = \boldsymbol{\nu}_{v_j \to c_p} \odot \boldsymbol{\nu}_{c_p \setminus v_\ell, v_j}$ as a suitable vector. Based on the fact that $\boldsymbol{\nu}_{c_p \setminus v_\ell, v_j}$ is a probability vector, together with the aforementioned component-wise ordering, we gather that

$$\begin{aligned}
\mathbf{0} \preceq \boldsymbol{\nu} &= \boldsymbol{\nu}_{v_j \to c_p} \odot \boldsymbol{\nu}_{c_p \setminus v_\ell, v_j} \\
&\preceq \boldsymbol{\mu}_{v_j \to c_p} \odot \boldsymbol{\nu}_{c_p \setminus v_\ell, v_j} = \boldsymbol{\mu}_{c_p \to v_\ell}.
\end{aligned}$$

Moreover, leveraging the properties of the $\mathbb{F}_q$ convolution for non-negative vectors, we get

$$\|\boldsymbol{\nu}\|_1 = \left\| \boldsymbol{\nu}_{v_j \to c_p} \right\|_1 \left\| \boldsymbol{\nu}_{c_p \setminus v_\ell, v_j} \right\|_1 = \left\| \boldsymbol{\nu}_{v_j \to c_p} \right\|_1.$$

Thus, for this choice of $\boldsymbol{\nu}$, we arrive at

$$\frac{\partial \boldsymbol{\mu}_{c_p \to v_\ell}}{\partial \mathbf{r}_j(h)} = \frac{1}{\tau^2} \left(\boldsymbol{\nu} - \|\boldsymbol{\nu}\|_1 \boldsymbol{\mu}_{c_p \to v_\ell} \right), \tag{66}$$

as claimed. That is, the base case conforms to the structure of Proposition 40.

We now consider the inductive step in our proof. As our hypothesis, we assume that (61) holds for all computation trees wherein the distance between the root node and v_j is less than or equal to $\gamma \in \mathbb{N}$. Consider a rooted computation tree and suppose the distance between v_ℓ and its descendant v_j in the tree is exactly $\gamma + 1$. Under Condition 9, there is a unique path from v_ℓ to node v_j. Let v_k be the variable child of v_ℓ that is also an ascendant of v_j, and denote the unique check node that connects the two by $c_p \in N(v_\ell) \cap N(v_k)$. Furthermore, let $c_\varrho \in N(v_k)$ be the unique check node within this neighborhood that is an ascendant of v_j on the computation tree. Finally, let $v_o \in N(c_\varrho)$ be the unique variable child of v_k that is also an ascendant of v_j (or, perhaps, v_j itself).

The sub-tree starting at v_k can be viewed as a rooted tree containing v_j; the graph distance between these two variable nodes within the sub-tree is exactly γ. As such, our inductive hypothesis applies. That is, there exists vector $\boldsymbol{\nu}$ such that $\mathbf{0} \preceq \boldsymbol{\nu} \preceq \boldsymbol{\mu}_{c_\varrho \to v_k}$ where the partial derivative of $\boldsymbol{\mu}_{c_\varrho \to v_k}$ with respect to $\mathbf{r}_j(h)$ is equal to

$$\frac{\partial \boldsymbol{\mu}_{c_\varrho \to v_k}}{\partial \mathbf{r}_j(h)} = \frac{1}{\tau^2} \left(\boldsymbol{\nu} - \|\boldsymbol{\nu}\|_1 \boldsymbol{\mu}_{c_\varrho \to v_k} \right). \tag{67}$$

Applying (62) and our inductive hypothesis, we have that

$$\frac{\partial \boldsymbol{\mu}_{v_k \to c_p}}{\partial \mathbf{r}_j(h)} = \frac{\partial}{\partial \mathbf{r}_j(h)} \frac{\bigcirc_{c_\xi \in N_0(v_k) \setminus c_p} \boldsymbol{\mu}_{c_\xi \to v_k}}{\left\| \bigcirc_{c_\xi \in N_0(v_k) \setminus c_p} \boldsymbol{\mu}_{c_\xi \to v_k} \right\|_1}$$

$$= \frac{\frac{\partial \boldsymbol{\mu}_{c_\varrho \to v_k}}{\partial \mathbf{r}_j(h)} \circ \left(\bigcirc_{c_\xi \in N_0(v_k) \setminus c_p, c_\varrho} \boldsymbol{\mu}_{c_\xi \to v_k} \right)}{\left\langle \boldsymbol{\mu}_{c_\varrho \to v_k}, \bigcirc_{c_\xi \in N_0(v_k) \setminus c_p, c_\varrho} \boldsymbol{\mu}_{c_\xi \to v_k} \right\rangle}$$
$$- \boldsymbol{\mu}_{v_k \to c_p} \frac{\left\langle \frac{\partial \boldsymbol{\mu}_{c_\varrho \to v_k}}{\partial \mathbf{r}_j(h)}, \bigcirc_{c_\xi \in N_0(v_k) \setminus c_p, c_\varrho} \boldsymbol{\mu}_{c_\xi \to v_k} \right\rangle}{\left\langle \boldsymbol{\mu}_{c_\varrho \to v_k}, \bigcirc_{c_\xi \in N_0(v_k) \setminus c_p, c_\varrho} \boldsymbol{\mu}_{c_\xi \to v_k} \right\rangle}$$
$$= \frac{1}{\tau^2} \frac{\boldsymbol{\nu} \circ \left(\bigcirc_{c_\xi \in N_0(v_k) \setminus c_p, c_\varrho} \boldsymbol{\mu}_{c_\xi \to v_k} \right)}{\left\langle \boldsymbol{\mu}_{c_\varrho \to v_k}, \bigcirc_{c_\xi \in N_0(v_k) \setminus c_p, c_\varrho} \boldsymbol{\mu}_{c_\xi \to v_k} \right\rangle}$$
$$- \frac{1}{\tau^2} \boldsymbol{\mu}_{v_k \to c_p} \frac{\left\langle \boldsymbol{\nu}, \bigcirc_{c_\xi \in N_0(v_k) \setminus c_p, c_\varrho} \boldsymbol{\mu}_{c_\xi \to v_k} \right\rangle}{\left\langle \boldsymbol{\mu}_{c_\varrho \to v_k}, \bigcirc_{c_\xi \in N_0(v_k) \setminus c_p, c_\varrho} \boldsymbol{\mu}_{c_\xi \to v_k} \right\rangle}$$
$$= \frac{1}{\tau^2} \left(\boldsymbol{\nu}_{v_k \to c_p} - \left\| \boldsymbol{\nu}_{v_k \to c_p} \right\|_1 \boldsymbol{\mu}_{v_k \to c_p} \right), \tag{68}$$

where we have utilized the shorthand notation

$$\boldsymbol{\nu}_{v_k \to c_p} = \frac{\boldsymbol{\nu} \circ \left(\bigcirc_{c_\xi \in N_0(v_k) \setminus c_p, c_\varrho} \boldsymbol{\mu}_{c_\xi \to v_k} \right)}{\left\langle \boldsymbol{\mu}_{c_\varrho \to v_k}, \bigcirc_{c_\xi \in N_0(v_k) \setminus c_p, c_\varrho} \boldsymbol{\mu}_{c_\xi \to v_k} \right\rangle}.$$

We emphasize that two of the terms in the derivation above cancel out, as before. Furthermore, by construction, we immediately obtain $\mathbf{0} \preceq \boldsymbol{\nu}_{v_k \to c_p} \preceq \boldsymbol{\mu}_{v_k \to c_p}$. These observations closely parallel the description for the base case.

The derivation of the second graph operation for the inductive step is in complete analogy with the base case, except for labeling. Specifically, we apply (63) and obtain

$$\frac{\partial \boldsymbol{\mu}_{c_p \to v_\ell}}{\partial \mathbf{r}_j(h)} = \frac{\partial \boldsymbol{\mu}_{v_k \to c_p}}{\partial \mathbf{r}_j(h)} \odot \boldsymbol{\nu}_{c_p \setminus v_\ell, v_k}$$
$$= \tfrac{1}{\tau^2} \left(\boldsymbol{\nu}_{v_k \to c_p} - \left\| \boldsymbol{\nu}_{v_k \to c_p} \right\|_1 \boldsymbol{\mu}_{v_k \to c_p} \right) \odot \boldsymbol{\nu}_{c_p \setminus v_\ell, v_k}$$
$$= \tfrac{1}{\tau^2} \left(\boldsymbol{\nu}_{v_k \to c_p} \odot \boldsymbol{\nu}_{c_p \setminus v_\ell, v_k} - \left\| \boldsymbol{\nu}_{v_k \to c_p} \right\|_1 \boldsymbol{\mu}_{c_p \to v_\ell} \right). \tag{69}$$

For the inductive step, we define $\boldsymbol{\nu}' = \boldsymbol{\nu}_{v_k \to c_p} \odot \boldsymbol{\nu}_{c_p \setminus v_\ell, v_k}$ as the candidate vector. Based on component-wise ordering, we can write

$$\mathbf{0} \preceq \boldsymbol{\nu}' = \boldsymbol{\nu}_{v_k \to c_p} \odot \boldsymbol{\nu}_{c_p \setminus v_\ell, v_k}$$
$$\preceq \boldsymbol{\mu}_{v_k \to c_p} \odot \boldsymbol{\nu}_{c_p \setminus v_\ell, v_k} = \boldsymbol{\mu}_{c_p \to v_\ell}.$$

As before, we have that

$$\|\boldsymbol{\nu}'\|_1 = \left\| \boldsymbol{\nu}_{v_k \to c_p} \right\|_1 \left\| \boldsymbol{\nu}_{c_p \setminus v_\ell, v_k} \right\|_1 = \left\| \boldsymbol{\nu}_{v_k \to c_p} \right\|_1.$$

Hence, candidate vector $\boldsymbol{\nu}'$ is such that $\mathbf{0} \preceq \boldsymbol{\nu}' \preceq \boldsymbol{\mu}_{c_p \to v_\ell}$ and

$$\frac{\partial \boldsymbol{\mu}_{c_p \to v_\ell}}{\partial \mathbf{r}_j(h)} = \frac{1}{\tau^2} \left(\boldsymbol{\nu}' - \|\boldsymbol{\nu}'\|_1 \boldsymbol{\mu}_{c_p \to v_\ell} \right). \tag{70}$$

This completes the proof for Proposition 40. ∎

We have nearly attained out goal of showing that the magnitudes of the entries in the Jacobian matrix of $\boldsymbol{\eta}(\mathbf{r})$ with respect to $\mathbf{r}$ are uniformly bounded. To achieve the desired result, it sufficies to connect the partial derivative of the incoming message with the partial derivative of state estimate $\hat{\mathbf{s}}_\ell(\mathbf{r}, g)$. This is accomplished below.

Proposition 41: Under Condition 9, the absolute value of the entries in the Jacobian are bounded by

$$\left| \frac{\partial \hat{\mathbf{s}}_\ell(\mathbf{r}, g)}{\partial \mathbf{r}_j(h)} \right| \leq \frac{1}{\tau^2} \quad \forall g, h \in \mathbb{F}_q, \ell, j \in [L], \tag{71}$$

where $\tau > 0$ represents the standard deviation of the effective observation.

Proof: When v_j does not appear in the rooted tree of v_ℓ, the partial derivative is equal to zero and the result immediately follows. Furthermore, when $v_\ell = v_j$, the result follows from Corollary 38. Thus, we can focus on the scenario wherein v_j is a descendant of v_ℓ.

Let c_p be the unique check node in $N(v_\ell)$ that lies on the path between v_ℓ and v_j. By Proposition 40, there exists vector $\boldsymbol{\nu}$, with $\mathbf{0} \preceq \boldsymbol{\nu} \preceq \boldsymbol{\mu}_{c_p \to v_\ell}$, such that the partial derivative of $\boldsymbol{\mu}_{c_p \to v_\ell}$ with respect to $\mathbf{r}_j(h)$ is given by

$$\frac{\partial \boldsymbol{\mu}_{c_p \to v_\ell}}{\partial \mathbf{r}_j(h)} = \frac{1}{\tau^2} \left(\boldsymbol{\nu} - \|\boldsymbol{\nu}\|_1 \boldsymbol{\mu}_{c_p \to v_\ell} \right). \tag{72}$$

Drawing an analogy to (68), we have

$$\frac{\partial \hat{\mathbf{s}}_\ell(\mathbf{r})}{\partial \mathbf{r}_j(h)} = \frac{\partial}{\partial \mathbf{r}_j(h)} \frac{\bigcirc_{c_\xi \in N_0(v_\ell)} \boldsymbol{\mu}_{c_\xi \to v_\ell}}{\left\| \bigcirc_{c_\xi \in N_0(v_\ell)} \boldsymbol{\mu}_{c_\xi \to v_\ell} \right\|_1}$$
$$= \frac{1}{\tau^2} \frac{\boldsymbol{\nu} \circ \left(\bigcirc_{c_\xi \in N_0(v_\ell) \setminus c_p} \boldsymbol{\mu}_{c_\xi \to v_\ell} \right)}{\left\langle \boldsymbol{\mu}_{c_p \to v_\ell}, \bigcirc_{c_\xi \in N_0(v_\ell) \setminus c_p} \boldsymbol{\mu}_{c_\xi \to v_\ell} \right\rangle}$$
$$- \frac{1}{\tau^2} \hat{\mathbf{s}}_\ell(\mathbf{r}) \frac{\left\langle \boldsymbol{\nu}, \bigcirc_{c_\xi \in N_0(v_\ell) \setminus c_p} \boldsymbol{\mu}_{c_\xi \to v_\ell} \right\rangle}{\left\langle \boldsymbol{\mu}_{c_p \to v_\ell}, \bigcirc_{c_\xi \in N_0(v_\ell) \setminus c_p} \boldsymbol{\mu}_{c_\xi \to v_\ell} \right\rangle}$$
$$= \frac{1}{\tau^2} \left(\boldsymbol{\nu}_{v_\ell} - \|\boldsymbol{\nu}_{v_\ell}\|_1 \hat{\mathbf{s}}_\ell(\mathbf{r}) \right), \tag{73}$$

where we have implicitly defined

$$\boldsymbol{\nu}_{v_\ell} = \frac{\boldsymbol{\nu} \circ \left(\bigcirc_{c_\xi \in N_0(v_\ell) \setminus c_p} \boldsymbol{\mu}_{c_\xi \to v_\ell} \right)}{\left\langle \boldsymbol{\mu}_{c_p \to v_\ell}, \bigcirc_{c_\xi \in N_0(v_\ell) \setminus c_p} \boldsymbol{\mu}_{c_\xi \to v_\ell} \right\rangle}.$$

We note that $\mathbf{0} \preceq \boldsymbol{\nu}_{v_\ell} \preceq \hat{\mathbf{s}}_\ell(\mathbf{r})$. Thus, we have that:

$$\frac{\partial \hat{\mathbf{s}}_\ell(\mathbf{r})}{\partial \mathbf{r}_j(h)} \leq \frac{1}{\tau^2} \boldsymbol{\nu}_{v_\ell} \leq \frac{1}{\tau^2} \hat{\mathbf{s}}_\ell(\mathbf{r}). \tag{74}$$

Since we are interested in bounding the absolute value of the partial derivatives, we also consider a lower bound.

$$\frac{\partial \hat{\mathbf{s}}_\ell(\mathbf{r})}{\partial \mathbf{r}_j(h)} \geq -\frac{1}{\tau^2} \|\boldsymbol{\nu}_{v_\ell}\|_1 \hat{\mathbf{s}}_\ell(\mathbf{r}) \geq -\frac{1}{\tau^2} \hat{\mathbf{s}}_\ell(\mathbf{r}). \tag{75}$$

Combining these two observations with the properties of probability vectors, we obtain the desired expression. ∎

Under Condition 9, Proposition 41 establishes an element-wise upper bound on the absolute value of the Jacobian of $\boldsymbol{\eta}(\mathbf{r})$ with respect to $\mathbf{r}$, which is a sufficient condition for $\boldsymbol{\eta}(\mathbf{r})$ to be Lipschitz continuous with respect to $\mathbf{r}$. Thus, we have shown that the BP denoiser is Lipschitz continuous for fixed qL. However, Theorem 11 requires the denoiser to be *uniformly* Lipschitz with respect to its ambient dimension. In other words, there must exist a finite Lipschitz constant which is independent of qL. As shown below, this Lipschitz constant exists under the additional constraints of Condition 10.

Proof: [Proof of Theorem 11] Let $\mathbf{J}$ denote the Jacobian matrix of $\boldsymbol{\eta}(\mathbf{r})$. By the generalization of the Mean-Value

Theorem to vector-valued functions [49], it follows that, for any two effective observations $\mathbf{r}_0, \mathbf{r}_1 \in \mathbb{R}^{qL}$,

$$\|\boldsymbol{\eta}(\mathbf{r}_0) - \boldsymbol{\eta}(\mathbf{r}_1)\| \leq \|\mathbf{J}\| \|\mathbf{r}_0 - \mathbf{r}_1\|, \tag{76}$$

where $\|\mathbf{J}\|$ denotes the operator norm $\|\mathbf{J}\| = \sup_{\|\mathbf{x}\|=1} \|\mathbf{J}\mathbf{x}\|$. To show that the BP denoiser is uniformly Lipschitz, it thus suffices to show that $\|\mathbf{J}\| < U_{\mathbf{J}} < \infty$ for some upper bound $U_{\mathbf{J}}$, regardless of the ambient dimension qL.

Under Condition 9, the computation graph of the BP denoiser forms a tree and, under Condition 10, each variable node root is connected to no more than

$$\begin{aligned}\Lambda &= \sum_{i=1}^{N_{\max}} \left(d^c_{\max} d^v_{\max}\right)^i \\ &= \frac{d^c_{\max} d^v_{\max} \left(1 - \left(d^c_{\max} d^v_{\max}\right)^{N_{\max}}\right)}{1 - d^c_{\max} d^v_{\max}}\end{aligned} \tag{77}$$

variable nodes in the tree. Thus, every row and column of $\mathbf{J}$ have no more than $\Lambda q_{\max}$ non-zero entries each. Crucially, under Condition 10, $\Lambda q_{\max} < \infty$. Let $\boldsymbol{\ell}_{i,j}$ be the indicator function for the $i,j$the entry of $\mathbf{J}$; that is

$$\boldsymbol{\ell}_{i,j} = \begin{cases} 0 & \mathbf{J}_{i,j} = 0 \\ 1 & \text{otherwise.} \end{cases} \tag{78}$$

Consider the ith entry of $\mathbf{J}\mathbf{x}$ for some vector $\mathbf{x}$ such that $\|\mathbf{x}\| < \infty$. Using the Cauchy–Schwarz inequality, we can write

$$\begin{aligned}\left((\mathbf{J}\mathbf{x})_i\right)^2 &= \left(\left\langle \mathbf{J}^{\mathrm{T}}_{i,:}, \mathbf{x}\right\rangle\right)^2 = \left(\left\langle \mathbf{J}^{\mathrm{T}}_{i,:}, \boldsymbol{\ell}^{\mathrm{T}}_{i,:} \circ \mathbf{x}\right\rangle\right)^2 \\ &\leq \|\mathbf{J}_{i,:}\|^2 \left\|\boldsymbol{\ell}^{\mathrm{T}}_{i,:} \circ \mathbf{x}\right\|^2 \\ &\leq \frac{\Lambda q_{\max}}{\tau^4} \left\|\boldsymbol{\ell}^{\mathrm{T}}_{i,:} \circ \mathbf{x}\right\|^2\end{aligned}$$

where the last inequality leverages both Proposition 40 and (77). It thus follows that, for any such $\mathbf{x}$,

$$\begin{aligned}\|\mathbf{J}\mathbf{x}\|^2 &\leq \frac{\Lambda q_{\max}}{\tau^4} \sum_{i=1}^{qL} \left\|\boldsymbol{\ell}^{\mathrm{T}}_{i,:} \circ \mathbf{x}\right\|^2 \\ &= \frac{\Lambda q_{\max}}{\tau^4} \sum_{i=1}^{qL} \sum_{j=1}^{qL} \boldsymbol{\ell}_{i,j} \mathbf{x}_j^2 \\ &= \frac{\Lambda q_{\max}}{\tau^4} \sum_{j=1}^{qL} \mathbf{x}_j^2 \sum_{i=1}^{qL} \boldsymbol{\ell}_{i,j} \\ &\leq \frac{\left(\Lambda q_{\max}\right)^2}{\tau^4} \sum_{j=1}^{qL} \mathbf{x}_j^2 = \frac{\left(\Lambda q_{\max}\right)^2}{\tau^4} \|\mathbf{x}\|^2 .\end{aligned} \tag{79}$$

Therefore, the operator norm of $\mathbf{J}$ is bounded by

$$\begin{aligned}\|\mathbf{J}\| &= \sup_{\|\mathbf{x}\|_2=1} \|\mathbf{J}\mathbf{x}\| \\ &\leq \sup_{\|\mathbf{x}\|_2=1} \frac{\Lambda q_{\max}}{\tau^2} \|\mathbf{x}\| \\ &= \frac{\Lambda q_{\max}}{\tau^2}\end{aligned} \tag{80}$$

which is finite and independent of both q and L. We thus conclude that the BP denoiser is uniformly Lipschitz. ■

B. Proof of Proposition 12

Intuitively, the role of the Onsager term is to (asymptotically) cancel the first-order correlations between $\mathbf{A}^{\mathrm{T}}\mathbf{z}^{(t)}$ and $\mathbf{s}^{(t)}$ and thereby maintain a structure conducive to prompt convergence and analysis. This factor, emblematic of AMP algorithms, appears in (6) and is given by

$$\begin{aligned}&\frac{\mathbf{z}^{(t-1)}}{n} \operatorname{div} \boldsymbol{\eta}_{t-1}\left(\mathbf{A}^{\mathrm{T}}\mathbf{z}^{(t-1)} + \mathbf{s}^{(t-1)}\right) 6 \\ &\qquad = \frac{\mathbf{z}^{(t-1)}}{n} \operatorname{div} \boldsymbol{\eta}_{t-1}\left(\mathbf{r}^{(t-1)}\right)\end{aligned} \tag{81}$$

where the div operator can be expanded into

$$\operatorname{div} \boldsymbol{\eta}\left(\mathbf{r}\right) = \sum_{\ell \in [L]} \operatorname{div} \hat{\mathbf{s}}_\ell(\mathbf{r}, \tau) = \sum_{\ell \in [L]} \sum_{g \in \mathbb{F}_q} \frac{\partial \hat{s}_\ell\left(g, \mathbf{r}, \tau\right)}{\partial \mathbf{r}_\ell(g)}. \tag{82}$$

Thus, as an intermediate step, we must calculate the partial derivative of $\hat{s}_\ell\left(g, \mathbf{r}, \tau\right)$ with respect to $\mathbf{r}_\ell(g)$. This computation is rendered much simpler under Condition 9, which ensures that the message passing operations employed during denoising yield valid computation trees without cycles.

Lemma 42: Under Condition 9, the partial derivative of $\hat{s}_\ell\left(g, \mathbf{r}, \tau\right)$ with respect to $\mathbf{r}_\ell(g)$ is equal to

$$\frac{\partial \hat{s}_\ell\left(g, \mathbf{r}, \tau\right)}{\partial \mathbf{r}_\ell(g)} = \frac{1}{\tau^2} \hat{s}_\ell\left(g, \mathbf{r}, \tau\right)\left(1 - \hat{s}_\ell\left(g, \mathbf{r}, \tau\right)\right) \tag{83}$$

where $g \in \mathbb{F}_q$.

Proof: Recall that the output of the BP denoiser defined in (27) can be expressed as

$$\begin{aligned}\hat{s}_\ell\left(g, \mathbf{r}, \tau\right) &= \frac{\boldsymbol{\alpha}_\ell(g) \prod_{c_p \in N(v_\ell)} \boldsymbol{\mu}_{c_p \to v_\ell}(g)}{\sum_{h \in \mathbb{F}_q} \boldsymbol{\alpha}_\ell(h) \prod_{c_p \in N(v_\ell)} \boldsymbol{\mu}_{c_p \to v_\ell}(h)} \\ &= \frac{\boldsymbol{\alpha}_\ell(g) \boldsymbol{\mu}_{v_\ell \to c_0}(g)}{\sum_{h \in \mathbb{F}_q} \boldsymbol{\alpha}_\ell(h) \boldsymbol{\mu}_{v_\ell \to c_0}(h)}.\end{aligned} \tag{84}$$

Under Condition 9, belief vector $\boldsymbol{\mu}_{v_\ell \to c_0}$ is based solely on extrinsic information and, hence, it is determined based on $\{\mathbf{r}_j : j \in [L] \setminus \{\ell\}\}$. Consequence, we gather that

$$\frac{\partial \boldsymbol{\mu}_{v_\ell \to c_0}(g)}{\partial \mathbf{r}_\ell(g)} = 0.$$

Under such circumstances, we can calculate the desired derivative in a straightforward manner, with

$$\begin{aligned}\frac{\partial \hat{s}_\ell\left(g, \mathbf{r}, \tau\right)}{\partial \mathbf{r}_\ell(g)} &= \frac{\partial}{\partial \mathbf{r}_\ell(g)} \frac{e^{\frac{\mathbf{r}_\ell(g)}{\tau^2}} \boldsymbol{\mu}_{v_\ell \to c_0}(g)}{\sum_{h \in \mathbb{F}_q} e^{\frac{\mathbf{r}_\ell(h)}{\tau^2}} \boldsymbol{\mu}_{v_\ell \to c_0}(h)} \\ &= \frac{1}{\tau^2} \frac{e^{\frac{\mathbf{r}_\ell(g)}{\tau^2}} \boldsymbol{\mu}_{v_\ell \to c_0}(g)}{\sum_{h \in \mathbb{F}_q} e^{\frac{\mathbf{r}_\ell(g)}{\tau^2}} \boldsymbol{\mu}_{v_\ell \to c_0}(g)} \\ &\quad - \frac{1}{\tau^2} \frac{\left(e^{\frac{\mathbf{r}_\ell(g)}{\tau^2}} \boldsymbol{\mu}_{v_\ell \to c_0}(g)\right)^2}{\left(\sum_{h \in \mathbb{F}_q} e^{\frac{\mathbf{r}_\ell(g)}{\tau^2}} \boldsymbol{\mu}_{v_\ell \to c_0}(g)\right)^2} \\ &= \frac{1}{\tau^2} \hat{s}_\ell\left(g, \mathbf{r}, \tau\right)\left(1 - \hat{s}_\ell\left(g, \mathbf{r}, \tau\right)\right).\end{aligned}$$

This last line corresponds to the statement of the lemma. ■

It is worth emphasizing that the derivative in (83) remains unchanged irrespective of the number of BP rounds computed

on the factor graph, so long as Condition 9 is satisfied. The divergence of (82) assumes the same simple form under such circumstances.

Proposition 43: The divergence of $\boldsymbol{\eta}(\mathbf{r})$ with respect to $\mathbf{r}$ is equal to

$$\operatorname{div}\boldsymbol{\eta}(\mathbf{r}) = \frac{1}{\tau^2}\left(\|\boldsymbol{\eta}(\mathbf{r})\|_1 - \|\boldsymbol{\eta}(\mathbf{r})\|^2\right). \tag{85}$$

Proof: We expand the div operator as

$$\begin{aligned}\operatorname{div}\boldsymbol{\eta}(\mathbf{r}) &= \sum_{\ell\in[L]}\operatorname{div}\hat{\mathbf{s}}_\ell(\mathbf{r},\tau) = \sum_{\ell\in[L]}\sum_{g\in\mathbb{F}_q}\frac{\partial\hat{s}_\ell(g,\mathbf{r},\tau)}{\partial\mathbf{r}_\ell(g)}\\ &= \sum_{\ell\in[L]}\sum_{g\in\mathbb{F}_q}\frac{1}{\tau^2}\hat{s}_\ell(g,\mathbf{r},\tau)\left(1-\hat{s}_\ell(g,\mathbf{r},\tau)\right)\\ &= \frac{1}{\tau^2}\left(\|\boldsymbol{\eta}(\mathbf{r})\|_1 - \|\boldsymbol{\eta}(\mathbf{r})\|^2\right). \end{aligned}\tag{86}$$

The last equality follows from the fact that, since $\hat{s}_\ell(g,\mathbf{r},\tau)$ lies between zero and one, the corresponding partial derivative with respect to $\mathbf{r}_\ell(g)$ found in Lemma 42 is always non-negative. ∎

The proof of Proposition 12 follows immediately from the definition of the Onsager term (81) and Proposition 43.

Appendix B
State Evolution

In this appendix, we provide proofs for the propositions, lemmas, corollaries, and theorems from Section IV.

A. Proofs From Section IV-A

Proof of Proposition 15:

Proof: Fix two points $\mathbf{s},\mathbf{s}'\in\mathcal{S}$. Since every point in $\mathcal{S}$ is obtained by indexing an LDPC codeword, there exists $\mathbf{v},\mathbf{v}'$ in LDPC codebook $\mathcal{V}$ such that $\mathbf{v}$ maps to $\mathbf{s}$ and, similarly, $\mathbf{v}'$ maps to $\mathbf{s}'$. Furthermore, since the outer LDPC code is a linear code, we have $\mathbf{u}=\mathbf{v}'-\mathbf{v}\in\mathcal{V}$; that is, $\mathbf{u}$ is also a valid codeword.

Consider the invertible translation $\boldsymbol{\upsilon}\mapsto\boldsymbol{\upsilon}\oplus\mathbf{u}$ in $\mathbb{F}_q^L$. Then, for any codeword $\boldsymbol{\upsilon}\in\mathcal{V}$, we get $\boldsymbol{\upsilon}\oplus\mathbf{u}\in\mathcal{V}$ because codebook $\mathcal{V}$ is closed under addition. Focusing on every section individually, we have the mapping $\upsilon_\ell\mapsto\upsilon_\ell\oplus u_\ell$. This action induces a bijection acting on $\mathbb{F}_q$. Likewise, this produces a permutation of basis vectors with

$$\mathbf{e}_{\upsilon_\ell}\mapsto\mathbf{e}_{\pi(\upsilon_\ell)}=\mathbf{e}_{\upsilon_\ell\oplus u_\ell}.$$

Based on this correspondence, we can define a permutation matrix $\Pi^{(\ell)}$ that acts on $\mathbb{R}^q$ for each $\ell\in[L]$. Then, aggregating these L permutation matrices, we get the isometry $\boldsymbol{\Pi}=\operatorname{diag}\left(\Pi^{(1)},\ldots,\Pi^{(L)}\right)$ which, by construction, maps $\mathbf{s}$ to $\mathbf{s}'$ while leaving $\mathcal{S}$ invariant. Since $\mathbf{s}$ and $\mathbf{s}'$ are arbitrary, we conclude that $\mathcal{S}$ is geometrically uniform, as claimed in the proposition. ∎

Proof of Proposition 16:

Proof: We already know that the 2-norm of $\mathbf{r}_\ell-\mathbf{e}_j$ is invariant to permutations in its vector argument. Then, under Corollary 4, we have

$$\begin{aligned}f_{\mathbf{R}_\ell|\mathbf{S}_\ell}(\mathbf{r}_\ell|\mathbf{e}_g) &= \frac{1}{(2\pi)^{\frac{q}{2}}\tau^q}\exp\left(-\frac{\|\mathbf{r}_\ell-\mathbf{e}_g\|^2}{2\tau^2}\right)\\ &= \frac{1}{(2\pi)^{\frac{q}{2}}\tau^q}\exp\left(-\frac{\|\Pi\mathbf{r}_\ell-\Pi\mathbf{e}_g\|^2}{2\tau^2}\right)\\ &= f_{\mathbf{R}_\ell|\mathbf{S}_\ell}(\Pi\mathbf{r}_\ell|\Pi\mathbf{e}_g).\end{aligned}\tag{87}$$

Leveraging the underlying order on field elements, we can write $\Pi\mathbf{e}_g=\mathbf{e}_{\pi(g)}$, where $\pi(\cdot)$ is the permutation on $\mathbb{F}_q$ induced by matrix Π. ∎

Proof of Corollary 17:

Proof: To begin, recall that $\mathbf{R}_\ell^{\times\omega}$ corresponds to the operator introduced in Definition 6. This operator reorders the entries of a vector based on the mapping $g\mapsto\omega\otimes g$, together with the bijection of Remark 1. This action therefore creates a permutation matrix Π on the vector entries of its argument. However, the zeroth element of $\mathbf{R}$ necessarily remains in its original location under this mapping because $\omega\otimes 0=0$. Thus, we can apply Proposition 16 and get

$$\begin{aligned}f_{\mathbf{R}_\ell|\mathbf{S}_\ell}(\mathbf{r}_\ell|\mathbf{e}_0) &= f_{\mathbf{R}_\ell|\mathbf{S}_\ell}(\Pi\mathbf{r}_\ell|\Pi\mathbf{e}_0)\\ &= f_{\mathbf{R}_\ell|\mathbf{S}_\ell}\left(\mathbf{r}_\ell^{\times\omega}\middle|\mathbf{e}_0\right).\end{aligned}\tag{88}$$

Consequently, the conditional distributions of $\mathbf{R}_\ell$ and $\mathbf{R}_\ell^{\times\omega}$ are identical, given vector input $\mathbf{e}_0$. ∎

B. Proofs From Section IV-B

Proof of Lemma 23:

Proof: This lemma is best proved by induction; however, in the interest of space, we only provide a proof of the base case. Let $\mathbf{L}$ and $\mathbf{M}$ be two independent group-symmetric likelihood-vector random variables. Define $\mathbf{N}=\mathbf{L}\odot\mathbf{M}$. Then, the components of the convolution are given by

$$N_g=(\mathbf{L}\odot\mathbf{M})_g=\sum_{h\in\mathbb{F}_q}L_hM_{g-h}.$$

As a function of two random variables with non-negative entries, it immediately follows that $\mathbf{N}$ is a likelihood vector random variable. What remains is to show that $\mathbf{N}$ is dominant and group-symmetric. For any $i\in\mathbb{F}_q\setminus\{0\}$, we have

$$\begin{aligned}\left(\mathbf{N}^{\times i}\right)_g &= \left((\mathbf{L}\odot\mathbf{M})^{\times i}\right)_g = (\mathbf{L}\odot\mathbf{M})_{g\otimes i}\\ &= \sum_{h\in\mathbb{F}_q}L_hM_{g\otimes i-h} = \sum_{h\in\mathbb{F}_q}L_{h\otimes i^{-1}\otimes i}M_{(g-h\otimes i^{-1})\otimes i}\\ &= \sum_{j\in\mathbb{F}_q}L_{j\otimes i}M_{(g-j)\otimes i} = \sum_{j\in\mathbb{F}_q}\left(\mathbf{L}^{\times i}\right)_j\left(\mathbf{M}^{\times i}\right)_{(g-j)}\\ &= \left(\mathbf{L}^{\times i}\odot\mathbf{M}^{\times i}\right)_g\end{aligned}\tag{89}$$

where $j=h\otimes i^{-1}$. By definition, $\mathbf{L}\overset{d}{=}\mathbf{L}^{\times i}$ and $\mathbf{M}\overset{d}{=}\mathbf{M}^{\times i}$, where $\overset{d}{=}$ denotes equality in distribution. Thus, we have that

$$\mathbf{N}=(\mathbf{L}\odot\mathbf{M})\overset{d}{=}\left(\mathbf{L}^{\times i}\odot\mathbf{M}^{\times i}\right)=\mathbf{N}^{\times i}.\tag{90}$$

Thus, $\mathbf{N}$ is a group symmetric likelihood vector random variable. Now, note that

$$\begin{aligned}\mathbb{E}[N_0] &= \mathbb{E}\left[\sum_{h\in\mathbb{F}_q}L_hM_{-h}\right] = \sum_{h\in\mathbb{F}_q}\mathbb{E}[L_h]\mathbb{E}[M_{-h}]\\ &= \mathbb{E}[L_0]\mathbb{E}[M_0]+(q-1)\mathbb{E}[L_\bullet]\mathbb{E}[M_\bullet]\end{aligned}$$

$$\geq \mathbb{E}[L_0]\mathbb{E}[M_\bullet] + \mathbb{E}[L_\bullet]\mathbb{E}[M_0] + (q-2)\mathbb{E}[L_\bullet]\mathbb{E}[M_\bullet]$$
$$= \mathbb{E}[N_\bullet].$$

The inequality above is a consequence of the fact that, by assumption, $\mathbf{L}$ and $\mathbf{M}$ are dominant symmetric. Hence, $\mathbb{E}[L_0] \geq \mathbb{E}[L_\bullet]$ and $\mathbb{E}[M_0] \geq \mathbb{E}[M_\bullet]$. Thus, $\mathbf{N}$ is a dominant group-symmetric likelihood vector random variable.

The extension of these findings to the convolution of multiple independent vectors follows from a straightforward induction argument. ∎

Proof of Lemma 24:

Proof: Again, in the interest of space, we only prove the base case when $\gamma = 2$. The inductive case for $\gamma > 2$ follows directly. Let $\boldsymbol{\ell}$ and $\mathbf{m}$ be likelihood-vectors over $\mathbb{F}_q$, and define $\mathbf{n} = \boldsymbol{\ell} \odot \mathbf{m}$. Since $\mathbb{F}_q$ is a finite set and likelihood-vectors have non-negative entries, we get

$$\begin{aligned}\|\mathbf{n}\|_1 &= \sum_{g\in\mathbb{F}_q} \mathbf{n}_g = \sum_{g\in\mathbb{F}_q} (\boldsymbol{\ell}\odot\mathbf{m})_g \\ &= \sum_{g\in\mathbb{F}_q}\sum_{h\in\mathbb{F}_q} \ell_h m_{g-h} = \sum_{h\in\mathbb{F}_q} \ell_h \sum_{g\in\mathbb{F}_q} m_{g-h} \\ &= \sum_{h\in\mathbb{F}_q} \ell_h \|\mathbf{m}\|_1 = \|\boldsymbol{\ell}\|_1 \|\mathbf{m}\|_1 . \end{aligned} \tag{91}$$

That is, the one-norm of $\mathbf{n}$ is equal to the product of the one-norms of $\boldsymbol{\ell}$ and $\mathbf{m}$. ∎

Proof of Corollary 25:

Proof: Let $\omega \in \Omega$ denote an outcome within the underlying sample space. For any such realization, Lemma 24 states that

$$\|\mathbf{N}(\omega)\|_1 = \prod_{p\in[\gamma]} \left\|\mathbf{L}^{(p)}(\omega)\right\|_1 .$$

Taking expectations with respect to the corresponding probability law, we get

$$\begin{aligned}\mathbb{E}\left[\|\mathbf{N}(\omega)\|_1\right] &= \mathbb{E}\left[\prod_{p\in[\gamma]} \left\|\mathbf{L}^{(p)}(\omega)\right\|_1\right] \\ &= \prod_{p\in[\gamma]} \mathbb{E}\left[\left\|\mathbf{L}^{(p)}(\omega)\right\|_1\right], \end{aligned} \tag{92}$$

where the last step is a consequence of the likelihood-vector random variables being independent. Hence, the expectations decouple and the result follows. ∎

Proof of Proposition 26:

Proof: Consider the case of two independent dominant group-symmetric likelihood-vector random variables, $\mathbf{L}$ and $\mathbf{M}$, with $\mathbf{N} = \mathbf{L} \odot \mathbf{M}$. Examining the zeroth component of $\mathbf{N}$, we write

$$\mathbb{E}\left[N_0\right] = \mathbb{E}[L_0]\mathbb{E}[M_0] + (q-1)\mathbb{E}[L_\bullet]\mathbb{E}[M_\bullet].$$

Similarly, the expected value of any other component within random vector $\mathbf{N}$ is of the form

$$\mathbb{E}\left[N_\bullet\right] = \mathbb{E}[L_0]\mathbb{E}[M_\bullet] + \mathbb{E}[L_\bullet]\mathbb{E}[M_0] + (q-2)\mathbb{E}[L_\bullet]\mathbb{E}[M_\bullet].$$

Combining these two equations, we arrive at the expression

$$\begin{aligned}\mathbb{E}\left[N_0\right] - \mathbb{E}\left[N_\bullet\right] &= \mathbb{E}[L_0]\mathbb{E}[M_0] - \mathbb{E}[L_0]\mathbb{E}[M_\bullet] \\ &\quad - \mathbb{E}[L_\bullet]\mathbb{E}[M_0] + \mathbb{E}[L_\bullet]\mathbb{E}[M_\bullet] \\ &= \left(\mathbb{E}[L_0] - \mathbb{E}[L_\bullet]\right)\left(\mathbb{E}[M_0] - \mathbb{E}[M_\bullet]\right), \end{aligned} \tag{93}$$

which seems propitious for the application of mathematical induction.

To proceed with the induction argument, we introduce a collection of convolved vectors: $\mathbf{N}^{(\gamma)} = \bigodot_{p\in[\gamma]} \mathbf{L}^{(p)}$. The hypothesis can be formulated as

$$\mathbb{E}\left[N_0^{(\gamma)}\right] - \mathbb{E}\left[N_\bullet^{(\gamma)}\right] = \prod_{p\in[\gamma]} \left(\mathbb{E}\left[L_0^{(p)}\right] - \mathbb{E}\left[L_\bullet^{(p)}\right]\right). \tag{94}$$

The base case of $\gamma = 1$ is immediate by construction. For the inductive step, assume that (94) holds for fixed $\gamma \in \mathbb{N}$, $\gamma > 0$. Note that

$$\mathbf{N}^{(\gamma+1)} = \mathbf{L}^{(\gamma+1)} \odot \mathbf{N}^{(\gamma)}.$$

Also, note that $\mathbf{L}^{(\gamma+1)}$ and $\mathbf{N}^{(\gamma)}$ are independent because $\left\{\mathbf{L}^{(p)}\right\}$ is a set of independent likelihood-vector random variables and, consequently, $\mathbf{L}^{(\gamma+1)}$ and $\left\{\mathbf{L}^{(p)} : p \in [\gamma]\right\}$ form independent collections. Hence, (93) applies and

$$\begin{aligned}&\mathbb{E}\left[N_0^{(\gamma+1)}\right] - \mathbb{E}\left[N_\bullet^{(\gamma+1)}\right] \\ &= \left(\mathbb{E}\left[L_0^{(\gamma+1)}\right] - \mathbb{E}\left[L_\bullet^{(\gamma+1)}\right]\right)\left(\mathbb{E}\left[N_0^{(\gamma)}\right] - \mathbb{E}\left[N_\bullet^{(\gamma)}\right]\right) \\ &= \left(\mathbb{E}\left[L_0^{(\gamma+1)}\right] - \mathbb{E}\left[L_\bullet^{(\gamma+1)}\right]\right) \prod_{p\in[\gamma]} \left(\mathbb{E}\left[L_0^{(p)}\right] - \mathbb{E}\left[L_\bullet^{(p)}\right]\right) \\ &= \prod_{p\in[\gamma+1]} \left(\mathbb{E}\left[L_0^{(p)}\right] - \mathbb{E}\left[L_\bullet^{(p)}\right]\right), \end{aligned}$$

where the penultimate equality follows from our inductive hypothesis. This completes the mathematical induction.

Lemma 23 asserts that $\mathbf{N} = \bigodot_{p\in[\gamma]} \mathbf{L}^{(p)}$ is a dominant group-symmetric likelihood-vector random variable. Thus,

$$\begin{aligned}\mathbb{E}\left[\|\mathbf{N}\|_1\right] &= \mathbb{E}\left[N_0\right] + (q-1)\mathbb{E}\left[N_\bullet\right] \\ &= \left(\mathbb{E}\left[N_0\right] - \mathbb{E}\left[N_\bullet\right]\right) + q\mathbb{E}\left[N_\bullet\right] \\ &= q\mathbb{E}\left[N_0\right] - (q-1)\left(\mathbb{E}\left[N_0\right] - \mathbb{E}\left[N_\bullet\right]\right). \end{aligned} \tag{95}$$

Furthermore, combining Corollary 25 and the fact that the random vectors in $\left\{\mathbf{L}^{(p)}\right\}$ are independent, we get

$$\mathbb{E}\left[\|\mathbf{N}\|_1\right] = \prod_{p\in[\gamma]} \left(\mathbb{E}\left[L_0^{(p)}\right] + (q-1)\mathbb{E}\left[L_\bullet^{(p)}\right]\right). \tag{96}$$

Isolating $\mathbb{E}\left[N_0\right]$ in (95) and then applying (94) & (96), we get

$$\begin{aligned}\mathbb{E}\left[N_0\right] &= \frac{1}{q}\mathbb{E}\left[\|\mathbf{N}\|_1\right] + \frac{(q-1)}{q}\left(\mathbb{E}\left[N_0\right] - \mathbb{E}\left[N_\bullet\right]\right) \\ &= \frac{1}{q}\prod_{p\in[\gamma]} \left(\mathbb{E}\left[L_0^{(p)}\right] + (q-1)\mathbb{E}\left[L_\bullet^{(p)}\right]\right) \\ &\quad + \left(\frac{q-1}{q}\right)\prod_{p\in[\gamma]} \left(\mathbb{E}\left[L_0^{(p)}\right] - \mathbb{E}\left[L_\bullet^{(p)}\right]\right). \end{aligned}$$

Similarly, isolating $\mathbb{E}\left[N_\bullet\right]$, we arrive at

$$\begin{aligned}\mathbb{E}[N_\bullet] &= \frac{1}{q}\mathbb{E}\left[\|\mathbf{N}\|_1\right] - \frac{1}{q}\left(\mathbb{E}[N_0] - \mathbb{E}[N_\bullet]\right) \\ &= \frac{1}{q}\prod_{p\in[\gamma]} \left(\mathbb{E}\left[L_0^{(p)}\right] + (q-1)\mathbb{E}\left[L_\bullet^{(p)}\right]\right) \end{aligned}$$

$$-\frac{1}{q}\prod_{p\in[\gamma]}\left(\mathbb{E}\left[L_0^{(p)}\right]-\mathbb{E}\left[L_\bullet^{(p)}\right]\right).$$

This completes the proof. ∎

Proof of Corollary 27:

Proof: First, we stress that a probability-vector random variable, as described in Definition 21, is also a likelihood-vector random variable, albeit with additional structure. Thus, some of the results derived above for likelihood-vector random variables readily apply in the current scenario. For instance, in view of Lemma 24, we have

$$\left\|\bigodot_{p\in[\gamma]}\bar{\mathbf{L}}^{(p)}\right\|_1=\prod_{p\in[\gamma]}\left\|\bar{\mathbf{L}}^{(p)}\right\|_1=1.$$

Thus, $\left\|\bar{\mathbf{N}}\right\|_1=1$ and, consequently, $\bar{\mathbf{N}}$ is a valid probability-vector random variable as suggested by our notation. Second, we emphasize that the expected value of all its components, except for the zeroth entry, are equal, Given that a probability-vector takes on values in the simplex, we can therefore write

$$\mathbb{E}\left[\bar{L}_\bullet\right]=\frac{1-\mathbb{E}\left[\bar{L}_0\right]}{q-1}.$$

The difference between $\mathbb{E}\left[\bar{L}_0\right]$ and $\mathbb{E}\left[\bar{L}_\bullet\right]$ is subject to

$$\begin{aligned}\mathbb{E}\left[\bar{L}_0\right]-\mathbb{E}\left[\bar{L}_\bullet\right]&=\mathbb{E}\left[\bar{L}_0\right]-\frac{1-\mathbb{E}\left[\bar{L}_0\right]}{q-1}\\&=\frac{q}{q-1}\left(\mathbb{E}\left[\bar{L}_0\right]-\frac{1}{q}\right).\end{aligned}$$

Collecting these findings and substituting the equivalent forms into Proposition 26, we arrive at the claimed expressions. ∎

Proof of Lemma 28:

Proof: Recall that, for a dominant permutation-symmetric Gaussian probability-vector random variable, we can write the individual components of $\bar{\mathbf{L}}$ as

$$\bar{L}_g=\frac{f_{\mathbf{R}_\ell|\mathbf{S}_\ell}\left(\mathbf{R}_\ell|\mathbf{e}_g\right)}{\sum_{h\in\mathbb{F}_q}f_{\mathbf{R}_\ell|\mathbf{S}_\ell}\left(\mathbf{R}_\ell|\mathbf{e}_h\right)}\tag{97}$$

where $f_{\mathbf{R}_\ell|\mathbf{S}_\ell}\left(\cdot|\mathbf{e}_g\right)$ is specified in (87). Interestingly, we can express the second moment of $\bar{L}_g$ as

$$\begin{aligned}\mathbb{E}\left[\bar{L}_g^2\right]&=\mathbb{E}_0\left[\left(\frac{f_{\mathbf{R}_\ell|\mathbf{S}_\ell}\left(\mathbf{R}_\ell|\mathbf{e}_g\right)}{\sum_{h\in\mathbb{F}_q}f_{\mathbf{R}_\ell|\mathbf{S}_\ell}\left(\mathbf{R}_\ell|\mathbf{e}_h\right)}\right)^2\right]\\&=\int_{\mathbb{R}^q}\frac{f_{\mathbf{R}_\ell|\mathbf{S}_\ell}\left(\mathbf{r}_\ell|\mathbf{e}_g\right)f_{\mathbf{R}_\ell|\mathbf{S}_\ell}\left(\mathbf{r}_\ell|\mathbf{e}_g\right)}{\left(\sum_{h\in\mathbb{F}_q}f_{\mathbf{R}_\ell|\mathbf{S}_\ell}\left(\mathbf{r}_\ell|\mathbf{e}_h\right)\right)^2}f_{\mathbf{R}_\ell|\mathbf{S}_\ell}\left(\mathbf{r}_\ell|\mathbf{e}_0\right)d\mathbf{r}_\ell\\&=\int_{\mathbb{R}^q}\frac{f_{\mathbf{R}_\ell|\mathbf{S}_\ell}\left(\mathbf{r}_\ell|\mathbf{e}_g\right)f_{\mathbf{R}_\ell|\mathbf{S}_\ell}\left(\mathbf{r}_\ell|\mathbf{e}_0\right)}{\left(\sum_{h\in\mathbb{F}_q}f_{\mathbf{R}_\ell|\mathbf{S}_\ell}\left(\mathbf{r}_\ell|\mathbf{e}_h\right)\right)^2}f_{\mathbf{R}_\ell|\mathbf{S}_\ell}\left(\mathbf{r}_\ell|\mathbf{e}_g\right)d\mathbf{r}_\ell\\&=\int_{\mathbb{R}^q}\frac{f_{\mathbf{R}_\ell|\mathbf{S}_\ell}\left(\mathbf{r}_\ell|\mathbf{e}_0\right)f_{\mathbf{R}_\ell|\mathbf{S}_\ell}\left(\mathbf{r}_\ell|\mathbf{e}_g\right)}{\left(\sum_{h\in\mathbb{F}_q}f_{\mathbf{R}_\ell|\mathbf{S}_\ell}\left(\mathbf{r}_\ell|\mathbf{e}_h\right)\right)^2}f_{\mathbf{R}_\ell|\mathbf{S}_\ell}\left(\mathbf{r}_\ell|\mathbf{e}_0\right)d\mathbf{r}_\ell\\&=\mathbb{E}\left[\bar{L}_0\bar{L}_g\right].\end{aligned}$$

In the third equality, we leverage the invariance in the problem structure established in Proposition 16. This fact, together with the symmetry in the region of integration, enables us to permute the indices. With this relation, we can rewrite the two-norm of $\bar{\mathbf{L}}$ as

$$\begin{aligned}\mathbb{E}\left[\left\|\bar{\mathbf{L}}\right\|_2^2\right]&=\mathbb{E}\left[\sum_{g\in\mathbb{F}_q}\bar{L}_g^2\right]=\mathbb{E}\left[\sum_{g\in\mathbb{F}_q}\bar{L}_0\bar{L}_g\right]\\&=\mathbb{E}\left[\bar{L}_0\left(\sum_{g\in\mathbb{F}_q}\bar{L}_g\right)\right]=\mathbb{E}\left[\bar{L}_0\left\|\bar{\mathbf{L}}\right\|_1\right]\\&=\mathbb{E}\left[\bar{L}_0\right].\end{aligned}\tag{98}$$

This chain of equalities reveals the intricate relation between $\mathbb{E}\left[\left\|\bar{\mathbf{L}}\right\|_2^2\right]$ and $\mathbb{E}\left[\bar{L}_0\right]$. ∎

Proof of Proposition 30:

Proof: First, suppose (46) holds. Then, we can write

$$\begin{aligned}\mathbb{E}\left[\left\|\bar{\mathbf{L}}\right\|_2^2\right]&=\mathbb{E}\left[\bar{L}_0\right]=\mathbb{E}\left[\bar{L}_0\left\|\bar{\mathbf{L}}\right\|_1\right]\\&=\mathbb{E}\left[\bar{L}_0\left(\bar{L}_0+(q-1)\bar{L}_\bullet\right)\right]\\&=\mathbb{E}\left[\bar{L}_0^2\right]+(q-1)\mathbb{E}\left[\bar{L}_0\bar{L}_\bullet\right].\end{aligned}$$

At the same time, by definition, we have

$$\mathbb{E}\left[\left\|\bar{\mathbf{L}}\right\|_2^2\right]=\mathbb{E}\left[\bar{L}_0^2\right]+(q-1)\mathbb{E}\left[\bar{L}_\bullet^2\right].$$

Equating both expressions for the two-norm of $\bar{\mathbf{L}}$, we deduce that $\mathbb{E}\left[\bar{L}_0\bar{L}_\bullet\right]=\mathbb{E}\left[\bar{L}_\bullet^2\right]$. To get the converse, we assume that $\bar{\mathbf{L}}$ is balanced and then parallel the progression in (98), which yields

$$\begin{aligned}\mathbb{E}\left[\left\|\bar{\mathbf{L}}\right\|_2^2\right]&=\mathbb{E}\left[\bar{L}_0^2\right]+(q-1)\mathbb{E}\left[\bar{L}_\bullet^2\right]\\&=\mathbb{E}\left[\bar{L}_0^2\right]+(q-1)\mathbb{E}\left[\bar{L}_0\bar{L}_\bullet\right]\\&=\mathbb{E}\left[\bar{L}_0\left(\bar{L}_0+(q-1)\bar{L}_\bullet\right)\right]=\mathbb{E}\left[\bar{L}_0\right].\end{aligned}$$

We emphasize that in the two instances above, we have leveraged the fact that, for any realization of $\bar{\mathbf{L}}$,

$$\left\|\bar{\mathbf{L}}\right\|_1=\bar{L}_0+(q-1)\bar{L}_\bullet=1.$$

Combining these two results, we get the desired logical equivalence. ∎

Proof of Corollary 31:

Proof: Let $\bar{\mathbf{L}}$ be a balanced dominant permutation-symmetric probability-vector random variable. Then, it follows that

$$\begin{aligned}\mathbb{E}\left[\left\|\bar{\mathbf{L}}-\mathbf{e}_0\right\|_2^2\right]&=\sum_{g\in\mathbb{F}_q}\mathbb{E}\left[\left(\bar{\mathbf{L}}_g-\mathbf{e}_0(g)\right)^2\right]\\&=\mathbb{E}\left[1-2\bar{\mathbf{L}}_0+\bar{\mathbf{L}}_0^2\right]+\sum_{g\in\mathbb{F}_q\setminus 0}\mathbb{E}\left[\bar{\mathbf{L}}_g^2\right]\\&=1-2\mathbb{E}\left[\bar{\mathbf{L}}_0\right]+\mathbb{E}\left[\left\|\bar{\mathbf{L}}\right\|_2^2\right]\\&=1-\mathbb{E}\left[\left\|\bar{\mathbf{L}}\right\|_2^2\right],\end{aligned}$$

where the last line uses the fact that $\mathbb{E}\left[\bar{\mathbf{L}}_0\right]=\mathbb{E}\left[\left\|\bar{\mathbf{L}}\right\|_2^2\right]$. ∎

Proof of Theorem 32:

Proof: We show this result via mathematical induction. To facilitate the proof, we need to expand our notation slightly with

$$\bar{\mathbf{N}}^{(\gamma)}=\bigodot_{p\in[\gamma]}\bar{\mathbf{L}}^{(p)}.$$

In this context, the base case is immediate. When $\gamma = 1$, we have $\bar{\mathbf{N}}^{(1)} = \bar{\mathbf{L}}^{(1)}$ and, based on our assumptions, we can write

$$\mathbb{E}\left[\left\|\bar{\mathbf{N}}^{(1)}\right\|_2^2\right] = \mathbb{E}\left[\left\|\bar{\mathbf{L}}^{(1)}\right\|_2^2\right] = \mathbb{E}\left[\bar{L}_0^{(1)}\right] = \mathbb{E}\left[\bar{N}_0^{(1)}\right].$$

For the inductive step, assume that (48) holds for γ fixed. Then, consider the case where

$$\bar{\mathbf{N}}^{(\gamma+1)} = \bar{\mathbf{L}}^{(\gamma+1)} \odot \bar{\mathbf{N}}^{(\gamma)}.$$

Within this part of the proof, we use the abridged notation $\bar{\mathbf{N}} = \bar{\mathbf{N}}^{(\gamma+1)}$, $\bar{\mathbf{L}} = \bar{\mathbf{L}}^{(\gamma+1)}$, and $\bar{\mathbf{M}} = \bar{\mathbf{N}}^{(\gamma)}$ to lighten the exposition. We stress that, under our inductive hypothesis, $\bar{\mathbf{L}}$ and $\bar{\mathbf{M}}$ are both balanced dominant permutation-symmetric probability-vector random variables. As a first step, we seek a convenient expression for the square of the two-norm of $\bar{\mathbf{N}}$,

$$\begin{aligned}
\mathbb{E}\left[\left\|\bar{\mathbf{N}}\right\|^2\right] &= \sum_{g\in\mathbb{F}_q} \mathbb{E}\left[\bar{N}_g^2\right] \\
&= \sum_{g\in\mathbb{F}_q} \mathbb{E}\left[\left(\sum_{h\in\mathbb{F}_q} \bar{L}_{g-h}\bar{M}_h\right)\left(\sum_{\iota\in\mathbb{F}_q} \bar{L}_{g-\iota}\bar{M}_\iota\right)\right] \\
&= \sum_{g\in\mathbb{F}_q}\sum_{h\in\mathbb{F}_q}\sum_{\iota\in\mathbb{F}_q} \mathbb{E}\left[\bar{L}_{g-h}\bar{L}_{g-\iota}\right]\mathbb{E}\left[\bar{M}_h\bar{M}_\iota\right] \\
&= \sum_{h\in\mathbb{F}_q} \mathbb{E}\left[\bar{M}_h\bar{M}_h\right] \sum_{g\in\mathbb{F}_q} \mathbb{E}\left[\bar{L}_{g-h}\bar{L}_{g-h}\right] \\
&\quad + \sum_{h\in\mathbb{F}_q}\sum_{\iota\in\mathbb{F}_q\setminus h} \mathbb{E}\left[\bar{M}_h\bar{M}_\iota\right] \sum_{g\in\mathbb{F}_q} \mathbb{E}\left[\bar{L}_{g-h}\bar{L}_{g-\iota}\right]. \qquad (99)
\end{aligned}$$

We emphasize that $\bar{\mathbf{L}}$ and $\bar{\mathbf{M}}$ are independent and, as such, we can split the expectations. Focusing on the first summand, we have

$$\begin{aligned}
&\sum_{h\in\mathbb{F}_q} \mathbb{E}\left[\bar{M}_h\bar{M}_h\right] \sum_{g\in\mathbb{F}_q} \mathbb{E}\left[\bar{L}_{g-h}\bar{L}_{g-h}\right] \\
&= \sum_{h\in\mathbb{F}_q} \mathbb{E}\left[\bar{M}_h^2\right] \sum_{g\in\mathbb{F}_q} \mathbb{E}\left[\bar{L}_{g-h}^2\right] = \sum_{h\in\mathbb{F}_q} \mathbb{E}\left[\bar{M}_h^2\right]\mathbb{E}\left[\left\|\bar{\mathbf{L}}\right\|_2^2\right] \\
&= \mathbb{E}\left[\left\|\bar{\mathbf{L}}\right\|_2^2\right]\mathbb{E}\left[\left\|\bar{\mathbf{M}}\right\|_2^2\right].
\end{aligned}$$

Turning to the second summand, we get

$$\begin{aligned}
&\sum_{h\in\mathbb{F}_q}\sum_{\iota\in\mathbb{F}_q\setminus h} \mathbb{E}\left[\bar{M}_h\bar{M}_\iota\right] \sum_{g\in\mathbb{F}_q} \mathbb{E}\left[\bar{L}_{g-h}\bar{L}_{g-\iota}\right] \\
&= \sum_{h\in\mathbb{F}_q}\sum_{\iota\in\mathbb{F}_q\setminus h} \mathbb{E}\left[\bar{M}_h\bar{M}_\iota\right]\left(2\mathbb{E}\left[\bar{L}_0\bar{L}_\bullet\right] + (q-2)\mathbb{E}\left[\bar{L}_\bullet\bar{L}_\star\right]\right) \\
&= \left(1 - \mathbb{E}\left[\left\|\bar{\mathbf{M}}\right\|_2^2\right]\right)\left(2\mathbb{E}\left[\bar{L}_0\bar{L}_\bullet\right] + (q-2)\mathbb{E}\left[\bar{L}_\bullet\bar{L}_\star\right]\right) \\
&= \left(1 - \mathbb{E}\left[\left\|\bar{\mathbf{M}}\right\|_2^2\right]\right)\left(2\mathbb{E}\left[\bar{L}_0\bar{L}_\bullet\right] + \mathbb{E}\left[\bar{L}_\bullet\right] - 2\mathbb{E}\left[\bar{L}_\bullet^2\right]\right) \\
&= \frac{1}{q-1}\left(1 - \mathbb{E}\left[\left\|\bar{\mathbf{L}}\right\|_2^2\right]\right)\left(1 - \mathbb{E}\left[\left\|\bar{\mathbf{M}}\right\|_2^2\right]\right).
\end{aligned}$$

The subscript notation $\bar{L}_\bullet\bar{L}_\star$ refers to any two distinct, non-zero elements in $\mathbb{F}_q$. In the second equality, we have utilized the fact that $\bar{\mathbf{M}}$ is normalized with $\left\|\bar{\mathbf{M}}\right\|_1 = 1$ and, hence,

$$1 - \left\|\bar{\mathbf{M}}\right\|_2^2 = \left\|\bar{\mathbf{M}}\right\|_1^2 - \left\|\bar{\mathbf{M}}\right\|_2^2 = \sum_{h\in\mathbb{F}_q}\sum_{\iota\in\mathbb{F}_q\setminus h} \bar{M}_h\bar{M}_\iota.$$

The third equality relies on the identity

$$\begin{aligned}
&\mathbb{E}\left[\bar{L}_\bullet\bar{L}_\star\right] = \mathbb{E}\left[\mathbb{E}\left[\bar{L}_\bullet\bar{L}_\star\middle|\bar{L}_0,\bar{L}_\bullet\right]\right] \\
&= \mathbb{E}\left[\bar{L}_\bullet\mathbb{E}\left[\bar{L}_\star\middle|\bar{L}_0,\bar{L}_\bullet\right]\right] = \mathbb{E}\left[\bar{L}_\bullet\left(\frac{1-\bar{L}_0-\bar{L}_\bullet}{q-2}\right)\right] \\
&= \frac{\mathbb{E}\left[\bar{L}_\bullet\right] - \mathbb{E}\left[\bar{L}_0\bar{L}_\bullet\right] - \mathbb{E}\left[\bar{L}_\bullet^2\right]}{q-2} = \frac{\mathbb{E}\left[\bar{L}_\bullet\right] - 2\mathbb{E}\left[\bar{L}_\bullet^2\right]}{q-2}.
\end{aligned}$$

The last equality makes use of the relations $\mathbb{E}\left[\bar{L}_0\bar{L}_\bullet\right] = \mathbb{E}\left[\bar{L}_\bullet^2\right]$ and $\mathbb{E}\left[\bar{L}_0\right] = \mathbb{E}\left[\left\|\bar{\mathbf{L}}\right\|_2^2\right]$, which hold for balanced dominant permutation-symmetric probability-vector random variables. Combining our findings for the constituent sums in (99), we arrive at

$$\begin{aligned}
&\mathbb{E}\left[\left\|\bar{\mathbf{N}}\right\|_2^2\right] \\
&= \mathbb{E}\left[\left\|\bar{\mathbf{L}}\right\|_2^2\right]\mathbb{E}\left[\left\|\bar{\mathbf{M}}\right\|_2^2\right] \\
&\quad + \frac{1}{q-1}\left(1 - \mathbb{E}\left[\left\|\bar{\mathbf{L}}\right\|_2^2\right]\right)\left(1 - \mathbb{E}\left[\left\|\bar{\mathbf{M}}\right\|_2^2\right]\right) \\
&= \frac{q}{q-1}\left(\mathbb{E}\left[\left\|\bar{\mathbf{L}}\right\|_2^2\right] - \frac{1}{q}\right)\left(\mathbb{E}\left[\left\|\bar{\mathbf{M}}\right\|_2^2\right] - \frac{1}{q}\right) + \frac{1}{q}.
\end{aligned}$$

This demonstrates that the inductive step is valid. Corollary 27 connects the expression for $\mathbb{E}\left[\left\|\bar{\mathbf{N}}\right\|_2^2\right]$ to the mean of $\bar{N}_0$, given the assumed condition

$$\mathbb{E}\left[\left\|\bar{\mathbf{L}}^{(p)}\right\|_2^2\right] = \mathbb{E}\left[\bar{L}_0^{(p)}\right] \qquad \forall p \in [\gamma].$$

Finally, Proposition 30 ensures that $\bar{\mathbf{N}}$ is balanced. This completes the proof. ∎

C. Proofs From Section IV-C

Proof of Proposition 33:

Proof: This result is obtained immediately by combining Theorem 32 and Corollary 31. ∎

Proof of Proposition 34:

Proof:

The operation at the variable node yields

$$\tilde{\boldsymbol{\mu}}_{v_\ell\to c_p} = \frac{\boldsymbol{\alpha}_\ell \circ \left(\bigcirc_{c_\xi\in N(v_\ell)\setminus c_p} \tilde{\boldsymbol{\mu}}_{c_\xi\to v_\ell}\right)}{\left\|\boldsymbol{\alpha}_\ell \circ \left(\bigcirc_{c_\xi\in N(v_\ell)\setminus c_p} \tilde{\boldsymbol{\mu}}_{c_\xi\to v_\ell}\right)\right\|_1}$$

or, component-wise,

$$\begin{aligned}
\tilde{\boldsymbol{\mu}}_{v_\ell\to c_p}(g) &= \frac{e^{\frac{\mathbf{r}_\ell(g)}{\tau^2}} \prod_{c_\xi\in N(v_\ell)\setminus c_p} \exp\left(\frac{\mathbf{r}_{\xi,\ell}(g)}{\tau_{\xi,\ell}^2}\right)}{\sum_{h\in\mathbb{F}_q}\left(e^{\frac{\mathbf{r}_\ell(h)}{\tau^2}} \prod_{c_\xi\in N(v_\ell)\setminus c_p} \exp\left(\frac{\mathbf{r}_{\xi,\ell}(h)}{\tau_{\xi,\ell}^2}\right)\right)} \\
&= \frac{\exp\left(\frac{\mathbf{r}_\ell(g)}{\tau^2} + \sum_{c_\xi\in N(v_\ell)\setminus c_p} \frac{\mathbf{r}_{\xi,\ell}(g)}{\tau_{\xi,\ell}^2}\right)}{\sum_{h\in\mathbb{F}_q} \exp\left(\frac{\mathbf{r}_\ell(h)}{\tau^2} + \sum_{c_\xi\in N(v_\ell)\setminus c_p} \frac{\mathbf{r}_{\xi,\ell}(h)}{\tau_{\xi,\ell}^2}\right)}.
\end{aligned}$$

We emphasize that the argument of the exponential is a Gaussian random variable with mean

$$\mathbb{E}\left[\frac{\mathbf{r}_\ell(g)}{\tau^2} + \sum_{\substack{c_\xi\in N(v_\ell)\\ c_\xi\neq c_p}} \frac{\mathbf{r}_{\xi,\ell}(g)}{\tau_{\xi,\ell}^2}\right] = \mathbf{s}_\ell(g)\left(\frac{1}{\tau^2} + \sum_{\substack{c_\xi\in N(v_\ell)\\ c_\xi\neq c_p}} \frac{1}{\tau_{\xi,\ell}^2}\right)$$

and variance

$$\operatorname{Var}\left[\frac{\mathbf{r}_\ell(g)}{\tau^2}+\sum_{\substack{c_\xi\in N(v_\ell)\\ c_\xi\neq c_p}}\frac{\mathbf{r}_{\xi,\ell}(g)}{\tau^2_{\xi,\ell}}\right]=\left(\frac{1}{\tau^2}+\sum_{\substack{c_\xi\in N(v_\ell)\\ c_\xi\neq c_p}}\frac{1}{\tau^2_{\xi,\ell}}\right).$$

Due to normalization, this becomes statistically equivalent to observing

$$\tilde{\mathbf{r}}_{\ell,p}=\mathbf{s}+\mathbf{n}_{\ell,p},$$

where $\mathbf{n}_{\ell,p}$ is i.i.d. with Gaussian entries

$$\mathcal{N}\left(0,\frac{1}{\frac{1}{\tau^2}+\sum_{c_\xi\in N(v_\ell)\setminus c_p}\frac{1}{\tau^2_{\xi,\ell}}}\right). \tag{100}$$

Denoting the variance of $\mathbf{n}_{\ell,p}$ by $\tilde{\tau}^2_{\ell,p}$, it follows that

$$\mathbb{E}\left[\tilde{\boldsymbol{\mu}}_{v_\ell\to c_p}(0)\right]=\Psi\left(\tilde{\tau}^2_{\ell,p}\right). \tag{101}$$

Since $\tilde{\boldsymbol{\mu}}_{v_\ell\to c_p}$ is a balanced dominant permutation-symmetric probability-vector random variable, the corresponding MSE may be obtained by inspection using Corollary 31. ■

REFERENCES

[1] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[2] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.

[3] M. Luby et al., "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inf. Theory.*, vol. 47, no. 2, pp. 585–598, Feb. 2001.

[4] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[5] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 657–670, Mar. 2001.

[6] M. C. Davey and D. J. C. MacKay, "Low density parity check codes over GF(q)," in *Proc. Inf. Theory Workshop*, 1998, pp. 70–71.

[7] A. Bennatan and D. Burshtein, "Design and analysis of nonbinary LDPC codes for arbitrary discrete-memoryless channels," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 549–583, Feb. 2006.

[8] T. J. Richardson and R. L. Urbanke, *Modern Coding Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[9] D. J. Costello, L. Dolecek, T. E. Fuja, J. Kliewer, D. G. M. Mitchell, and R. Smarandache, "Spatially coupled sparse codes on graphs: Theory and practice," *IEEE Commun. Mag.*, vol. 52, no. 7, pp. 168–176, Jul. 2014.

[10] A. J. Felstrom and K. S. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2181–2191, Sep. 1999.

[11] M. Lentmaier, A. Sridharan, D. J. Costello, and K. S. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5274–5289, Oct. 2010.

[12] S. Kudekar, T. Richardson, and R. L. Urbanke, "Spatially coupled ensembles universally achieve capacity under belief propagation," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 7761–7813, Dec. 2013.

[13] A. Yedla, Y.-Y. Jian, P. S. Nguyen, and H. D. Pfister, "A simple proof of Maxwell saturation for coupled scalar recursions," *IEEE Trans. Inf. Theory*, vol. 60, no. 11, pp. 6943–6965, Nov. 2014.

[14] S. Kumar, A. J. Young, N. Macris, and H. D. Pfister, "Threshold saturation for spatially coupled LDPC and LDGM codes on BMS channels," *IEEE Trans. Inf. Theory*, vol. 60, no. 12, pp. 7389–7415, Dec. 2014.

[15] I. Andriyanova and A. Graell I Amat, "Threshold saturation for nonbinary SC-LDPC codes on the binary erasure channel," *IEEE Trans. Inf. Theory*, vol. 62, no. 5, pp. 2622–2638, May 2016.

[16] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary LDPC codes over GF(q)," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 633–643, Apr. 2007.

[17] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard, "Low-complexity decoding for non-binary LDPC codes in high order fields," *IEEE Trans. Commun.*, vol. 58, no. 5, pp. 1365–1375, May 2010.

[18] B.-Y. Chang, D. Divsalar, and L. Dolecek, "Non-binary protograph-based LDPC codes for short block-lengths," in *Proc. IEEE Inf. Theory Workshop*, Sep. 2012, pp. 282–286.

[19] A. Joseph and A. R. Barron, "Least squares superposition codes of moderate dictionary size are reliable at rates up to capacity," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 2541–2557, May 2012.

[20] A. Joseph and A. R. Barron, "Fast sparse superposition codes have near exponential error probability for $R < C$," *IEEE Trans. Inf. Theory*, vol. 60, no. 2, pp. 919–942, Feb. 2014.

[21] R. Venkataramanan, S. Tatikonda, and A. Barron, "Sparse regression codes," *Found. Trends Commun. Inf. Theory*, vol. 15, nos. 1–2, pp. 1–195, 2019.

[22] J. Barbier and F. Krzakala, "Replica analysis and approximate message passing decoder for superposition codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2014, pp. 1494–1498.

[23] J. Barbier and F. Krzakala, "Approximate message-passing decoder and capacity achieving sparse superposition codes," *IEEE Trans. Inf. Theory*, vol. 63, no. 8, pp. 4894–4927, Aug. 2017.

[24] C. Rush, A. Greig, and R. Venkataramanan, "Capacity-achieving sparse superposition codes via approximate message passing decoding," *IEEE Trans. Inf. Theory*, vol. 63, no. 3, pp. 1476–1500, Mar. 2017.

[25] C. Rush, K. Hsieh, and R. Venkataramanan, "Capacity-achieving spatially coupled sparse superposition codes with AMP decoding," *IEEE Trans. Inf. Theory*, vol. 67, no. 7, pp. 4446–4484, Jul. 2021.

[26] A. Greig and R. Venkataramanan, "Techniques for improving the finite length performance of sparse superposition codes," *IEEE Trans. Commun.*, vol. 66, no. 3, pp. 905–917, Mar. 2018.

[27] H. Cao and P. O. Vontobel, "Using list decoding to improve the finite-length performance of sparse regression codes," *IEEE Trans. Commun.*, vol. 69, no. 7, pp. 4282–4293, Jul. 2021.

[28] S. Liang, C. Liang, J. Ma, and L. Ping, "Compressed coding, AMP-based decoding, and analog spatial coupling," *IEEE Trans. Commun.*, vol. 68, no. 12, pp. 7362–7375, Dec. 2020.

[29] A. Fengler, P. Jung, and G. Caire, "SPARCs for unsourced random access," *IEEE Trans. Inf. Theory*, vol. 67, no. 10, pp. 6894–6915, Oct. 2021.

[30] V. K. Amalladinne, A. K. Pradhan, C. Rush, J. Chamberland, and K. R. Narayanan, "Unsourced random access with coded compressed sensing: Integrating AMP and belief propagation," *IEEE Trans. Inf. Theory*, vol. 68, no. 4, pp. 2384–2409, Apr. 2022.

[31] J. R. Ebert, V. K. Amalladinne, S. Rini, J. Chamberland, and K. R. Narayanan, "Coded demixing for unsourced random access," *IEEE Trans. Signal Process.*, vol. 70, pp. 2972–2984, 2022.

[32] L. Liu, C. Liang, J. Ma, and L. Ping, "Capacity optimality of AMP in coded systems," *IEEE Trans. Inf. Theory*, vol. 67, no. 7, pp. 4429–4445, Jul. 2021.

[33] J. R. Ebert, J.-F. Chamberland, and K. R. Narayanan, "On sparse regression LDPC codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, May 2023, pp. 2350–2355.

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

[35] R. Berthier, A. Montanari, and P.-M. Nguyen, "State evolution for approximate message passing with non-separable functions," *Inf. Inference, J. IMA*, vol. 9, no. 1, pp. 33–79, Mar. 2020.

[36] C. Gerbelot and R. Berthier, "Graph-based approximate message passing iterations," *Inf. Inference, A J. IMA*, vol. 12, no. 4, pp. 2562–2628, Sep. 2023.

[37] 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.

[38] N. Wiberg, H.-A. Loeliger, and R. Kotter, "Codes and iterative decoding on general graphs," *Eur. Trans. Telecommun.*, vol. 6, no. 5, pp. 513–525, Sep. 1995.

[39] 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, Feb. 2001.

[40] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 28–41, Jan. 2004.

[41] R. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vols. IT-27, no. 5, pp. 533–547, Sep. 1981.
[42] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Efficient erasure correcting codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 569–584, Feb. 2001.
[43] H. Song and J. R. Cruz, "Reduced-complexity decoding of Q-ary LDPC codes for magnetic recording," *IEEE Trans. Magn.*, vol. 39, no. 2, pp. 1081–1087, Mar. 2003.
[44] A. Goupil, M. Colas, G. Gelle, and D. Declercq, "FFT-based BP decoding of general LDPC codes over Abelian groups," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 644–649, Apr. 2007.
[45] G. D. Forney, "Geometrically uniform codes," *IEEE Trans. Inf. Theory*, vol. 37, no. 5, pp. 1241–1260, Sep. 1991.
[46] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.
[47] G. J. Byers and F. Takawira, "EXIT charts for non-binary LDPC codes," in *Proc. IEEE Int. Conf. Commun., ICC*, vol. 1, May 2005, pp. 652–657.
[48] G. D. Forney and G. Ungerboeck, "Modulation and coding for linear Gaussian channels," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2384–2415, May 1998.
[49] W. S. Hall and M. L. Newell, "The mean value theorem for vector valued functions: A simple proof," *Math. Mag.*, vol. 52, no. 3, p. 157, May 1979.

Jamison R. Ebert (Member, IEEE) received the B.S. degree in electrical engineering from Brigham Young University, Provo, UT, USA, in 2020, and the Ph.D. degree in electrical engineering from Texas A&M University, College Station, TX, USA, in 2024. He is currently a Senior Engineer with Qualcomm Technologies, San Diego, CA, USA. His research interests include signal processing, wireless communications, error control coding, and compressed sensing.

Jean-Francois Chamberland (Senior Member, IEEE) received the B.Eng. degree from McGill University, the M.S. degree from Cornell University, and the Ph.D. degree from the University of Illinois at Urbana–Champaign. He is currently a Professor with the Department of Electrical and Computer Engineering, Texas A&M University. Recently, he has been studying the efficient design of wireless systems and the fundamental limits of communication networks. His research interests include information theory, statistical inference, algorithms, and learning. His contributions have been recognized through the IEEE Young Author Best Paper Award from the IEEE Signal Processing Society, the Faculty Early Career Development (CAREER) Award from the National Science Foundation (NSF), and the IEEE Communications Society and Information Theory Society Joint Paper Award. He serves as a Senior Area Editor for the IEEE OPEN JOURNAL OF SIGNAL PROCESSING. He was an Associate Editor of IEEE TRANSACTIONS ON INFORMATION THEORY.

Krishna R. Narayanan (Fellow, IEEE) received the B.E. degree from Coimbatore Institute of Technology, the M.S. degree from Iowa State University, and the Ph.D. degree in electrical engineering from Georgia Institute of Technology in 1998. Since 1998, he has been with the Department of Electrical and Computer Engineering, Texas A&M University, where he is currently the Sanchez Chair Professor. His current research interests include the design of massive uncoordinated multiple access schemes, coded distributed computing, and applications of machine learning to wireless communications. He was elected as a fellow of the IEEE for contributions to coding for wireless communications and data storage. He received the 2022 IEEE Joint Communications Society and Information Theory Society Best Paper Award. He also received the 2006 and 2020 Best Paper Awards from the IEEE Technical Committee for Signal Processing for Data Storage. He has served as an Associate Editor for Coding Techniques for IEEE TRANSACTIONS ON INFORMATION THEORY and as an Area Editor (and Editor) for the Coding Theory and Applications Area for IEEE TRANSACTIONS ON COMMUNICATIONS.