,(((,QWHUQDWLRQDO6PSRVLPRQ,QIRUPDWLRQ7KHRU,6,7

On Sparse Regression LDPC Codes

Jamison R. Ebert, Jean-Francois Chamberland, Krishna R. Narayanan
Department of Electrical and Computer Engineering, Texas A&M University

2023 IEEE International Symposium on Information Theory (ISIT) | 978-1-6654-7554-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISIT54713.2023.10206818

***Abstract*—Iterative decoding of graph-based codes and sparse recovery through approximate message passing (AMP) are two research areas that have seen monumental progress in recent decades. Inspired by these advances, this article introduces sparse regression LDPC codes (SR-LDPC codes) and their decoding. Sparse regression codes (SPARCs) are a class of error correcting codes that build on ideas from compressed sensing and can be decoded using AMP. In certain settings, SPARCs are known to achieve capacity; yet, their performance suffers at finite block lengths. Likewise, low-density parity-check (LDPC) codes can be decoded efficiently using belief propagation and can also be capacity achieving. This article introduces a novel concatenated coding structure that combines an LDPC outer code with a SPARC-inspired inner code. Efficient decoding for such a code can be achieved using AMP with a denoiser that performs belief propagation on the factor graph of the outer LDPC code. The proposed framework exhibits performance improvements over SPARCs and standard LDPC codes for finite block lengths and results in a steep waterfall in error performance, a phenomenon not observed in uncoded SPARCs.**

***Index Terms*—LDPC codes, sparse regression codes, approximate message passing, belief propagation.**

I. Introduction and Background

Low-density parity check (LDPC) codes have been the subject of many scientific inquiries in the past, with landmark contributions ranging from theoretical advances to practical implementations [1]–[8]. LDPC ensembles are amenable to analysis and, under certain conditions, encoded messages can be recovered efficiently using iterative belief propagation (BP) decoding. Interestingly, the complexity per iteration of BP decoding grows linearly with block length and, as such, this paradigm offers a pragmatic solution for long block lengths [9]. Some spatially coupled LDPC constructions feature capacity approaching iterative decoding thresholds while also avoiding the pitfall of error floors [10]–[15]. Unfortunately, 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 a means to provide adequate performance [6], [7], [16]–[18].

In a seemingly unrelated research direction, Joseph and Barron introduced the concept of a sparse regression code (SPARC), which establishes a connection between code design and sparse recovery in high dimensions [19]–[21]. The codewords contained in a SPARC codebook consist of sparse superpositions of columns from a design matrix, also known as a sensing matrix in the compressed sensing (CS) literature [22], [23]. In some sense, the SPARC decoding process is equivalent to recovering the non-zero entries in a suitably designed index vector. Low complexity algorithms have been proposed for the efficient decoding of SPARC codewords. Notably, Barbier *et al.* introduce an approximate message passing (AMP) [24]–[28] decoder for SPARCs in [29] which is further considered in [30], [31]. Sparse regression codes paired with AMP decoding are known to achieve asymptotic single-user AWGN channel capacity under an appropriately chosen power allocation. Concurrently, there have been efforts to improve the finite block-length performance of SPARCs [32], [33].

One strategy to improve the performance of codes in practical settings is to adopt a concatenated structure. For example, SPARC and LDPC codes have been paired in the past to create concatenated schemes that improve performance [32]–[35]. Concatenated structures with a SPARC-like inner code have also been proposed in unsourced random access [36]–[38]. In [37], it is shown that the structure of a judiciously designed outer code can be integrated into the denoising function of the inner AMP recovery algorithm. Surprisingly, despite being mentioned by Liu *et al.* in [35] as a possible future research direction, such an approach has not been considered for the single-user scenario. This article seeks to address this defficiency by introducing SR-LDPC codes.

SR-LDPC codes feature a concatenated inner SPARC-like code and an outer non-binary LDPC code. This concatenated structure is amenable to a decoder that exchanges information between inner and outer codes via a dynamic AMP denoiser. In this setting, the denoiser performs BP on the factor graph of the outer LDPC code to improve the AMP state estimates. This in turn improves the effective observations of AMP which translate into better local belief vectors for the BP LDPC decoder. Once the MSE of the iterative AMP-BP decoding levels off, the outer LDPC code can be leveraged once more to further improve performance. The goal is to use AMP-BP decoding to increase the SNR of the effective observation to a level that exceeds the BP threshold of the graph and then to apply message passing to the outer LDPC code to recover the message. Numerical results demonstrate the performance advantages of the proposed model over related architectures at reasonably short block lengths.

II. System Model

We consider a point-to-point memoryless additive white Gaussian noise (AWGN) channel with one transmitter and one receiver, each equipped with a single antenna. The received signal $\mathbf{y} \in \mathbb{R}^n$ is given by

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

This material is based upon work supported, in part, by the National Science Foundation (NSF) under Grant CCF-2131106, and by Qualcomm Technologies, Inc., through their University Relations Program.

k,(((
Authorized licensed use limited to: Texas A M University. Downloaded on September 30,2024 at 11:53:43 UTC from IEEE Xplore. Restrictions apply.

where $\mathbf{x} \in \mathbb{R}^n$ denotes the transmitted signal and $\mathbf{z} \sim \mathcal{N}\left(0, \sigma^2 \mathbf{I}\right)$ represents AWGN. As mentioned above, we wish to study a coding architecture composed of a sparse regression inner code [19]–[21], and a non-binary LDPC outer code [6]–[8]. We elaborate on the encoding and decoding processes below.

A. Sparse Regression LDPC Encoding

The proposed encoding process features a sequence of three distinct steps: q-ary LDPC encoding, indexing LDPC symbols, and inner CS encoding. First, the information bits are encoded into a q-ary LDPC codeword via well-established operations [6]–[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; the output is sometimes referred to as a large random matrix system [19]–[21], [35]. 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 $\mathbf{v} \in \mathbb{F}_q^L$; 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

$$\mathbf{v} = (v_1, v_2, \ldots, v_L) \tag{2}$$

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

Note that 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 $0 \in \mathbb{Z}$ maps to $0 \in \mathbb{F}_q$ and $1 \in \mathbb{Z}$ maps to $1 \in \mathbb{F}_q$. Throughout, we adopt such an arbitrary, but fixed bijection and 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 exposition, and its use should not lead to confusion because one can unambiguously infer from context whether an instance refers to the field element or to its integer representation.

2) LDPC Symbol Indexing: 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 seize this opportunity to reinforce the notion that entry ℓ of $\mathbf{v}$ is an element of $\mathbb{F}_q$, 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

$$\mathbf{s} = \begin{bmatrix} \mathbf{e}_{v_1} \\ \vdots \\ \mathbf{e}_{v_L} \end{bmatrix}, \tag{3}$$

where $\mathbf{s}$ is an L-sparse vector of length qL. Vector $\mathbf{s}$ has a structure akin to that of a sparse regression codeword 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 get $\mathbf{x} = \mathbf{A}\mathbf{s}$, where $\mathbf{A} \in \mathbb{R}^{n \times qL}$ and $\mathbf{A}_{i,j} \sim \mathcal{N}\left(0, \frac{1}{n}\right)$. Note that $n \ll qL$ to conform to the CS framework. Equation (1) may thus be rewritten as:

$$\mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{z}. \tag{4}$$

Having specified the code structure, we are now ready to discuss the process of recovering $\mathbf{s}$ from $\mathbf{y}$.

B. Sparse Regression LDPC Decoding

Paralleling the development of AMP for sparse regression codes [31] and drawing inspiration from concatenated AMP systems [34], [35], we wish to create a composite iterative process to recover state vector $\mathbf{s}$ from $\mathbf{y}$ that exploits the sparsity in $\mathbf{s}$ and the parity structure of the LDPC code. This can be accomplished by incorporating message passing on the factor graph of the LDPC code into the AMP denoiser. A similar approach can be found in [37]; however, the denoiser we wish to utilize below differs in the fact that only one codeword is present within $\mathbf{y}$.

Our AMP composite algorithm is as follows,

$$\mathbf{z}^{(t)} = \mathbf{y} - \mathbf{A}\mathbf{s}^{(t)} + \frac{\mathbf{z}^{(t-1)}}{n} \operatorname{div} \boldsymbol{\eta}_{t-1}\left(\mathbf{r}^{(t-1)}\right) \tag{5}$$

$$\mathbf{r}^{(t)} = \mathbf{A}^{\mathrm{T}}\mathbf{z}^{(t)} + \mathbf{s}^{(t)} \tag{6}$$

$$\mathbf{s}^{(t+1)} = \boldsymbol{\eta}_t\left(\mathbf{r}^{(t)}\right) \tag{7}$$

where the superscript t denotes the iteration count and div denotes the divergence operator from vector calculus. The algorithm is initialized with conditions $\mathbf{r}^{(0)} = \mathbf{s}^{(0)} = \mathbf{0}$ and $\mathbf{z}^{(0)} = \mathbf{y}$.

Equation (5) computes the *residual* error enhanced with an Onsager correction term. Equation (6) specifies the *effective observation*, which is characteristic of AMP, and (7) provides a *state estimate*. The denoising functions $(\boldsymbol{\eta}_t(\cdot))_{t \geq 0}$ seek to exploit the structure of $\mathbf{s}$ while computing the state updates.

Generally, one would employ the Bayes-optimal denoiser, which is the conditional expectation of $\mathbf{s}$ given $\mathbf{r}^{(t)}$. However, this approach is computationally intractable in the present context because it entails summing over all the possible codewords. As an alternative, we know that BP can be efficiently applied to q-ary LDPC codes. Furthermore, at any point during BP, a belief on individual LDPC symbols can be formed based on incoming messages from neighboring factor nodes and local observations. Thus, we can potentially apply a few iterations of BP as a means to get an estimate for the state vector by leveraging the connection between sections of $\mathbf{s}$ and LDPC symbols. In this sense, iterative message passing can act as a foundation for pragmatic denoising functions. We elaborate on this connection below and, concurrently, we review pertinent notions of BP applied to q-ary LDPC codes.

III. Creating a BP Denoiser for AMP

In this section, we introduce the denoising function we wish to employ within AMP. To begin, we emphasize that $\mathbf{r}$ admits

Authorized licensed use limited to: Texas A M University. Downloaded on September 30,2024 at 11:53:43 UTC from IEEE Xplore. Restrictions apply.

a sectionized representation akin to that of the state vector $\mathbf{s}$ in (3). That is, we can view both the state estimate and the effective observation as concatenations of L vectors, each of length q. Mathematically, we have

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

This point is crucially important because the denoiser is constructed in a block-wise fashion. As a side note, we neglect the superscript (t) for most of the discussion below to lighten notation; instead, we employ the hat symbol to distinguish the estimate $\hat{\mathbf{s}}$ from the true state vector $\mathbf{s}$.

Each section $\mathbf{r}_\ell$ in $\mathbf{r}$ acts as a vector observation about the value of $\mathbf{s}_\ell \in \{\mathbf{e}_g : g \in [q]\}$. An astounding and enabling property of AMP is that, under certain technical conditions, the effective observation $\mathbf{r}$ is asymptotically distributed as $\mathbf{s} + \tau\boldsymbol{\zeta}$, where $\boldsymbol{\zeta}$ is a random vector with independent $\mathcal{N}(0,1)$ components and τ is a deterministic quantity. This fact hinges on the presence of the Onsager term in (5) and on some smoothness conditions for the denoising functions. Due to space constraints, the treatment of these technical conditions is deferred to an extended version of this article [39].

Condition 1. *The effective observation $\mathbf{r}^{(t)}$ is asymptotically distributed as $\mathbf{s} + \tau_t\boldsymbol{\zeta}_t$, where $\boldsymbol{\zeta}_t$ is a random vector with independent $\mathcal{N}(0,1)$ components and τ_t is specified by a set of deterministic equations. This asymptotic characterization takes place in the dimensions of the system, as opposed to time or iteration count.*

We describe below our rationale behind the denoising function assuming Condition 1 holds; we provide a rigorous foundation for this condition in [39], Appendix B, but this can only be done once the structure of the denoiser is established. Consider the effective observation restricted to section ℓ. Under Condition 1, the distribution of random observation vector $\mathbf{R}_\ell$ given section $\mathbf{S}_\ell = \mathbf{e}_g$ is given by

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

Under a uniform input distribution, the conditional probability of $\mathbf{S}_\ell = \mathbf{e}_g$ becomes

$$\begin{aligned} \boldsymbol{\alpha}_\ell(g) &= \Pr(\mathbf{S}_\ell = \mathbf{e}_g|\mathbf{R}_\ell = \mathbf{r}_\ell) = \Pr(V_\ell = g|\mathbf{R}_\ell = \mathbf{r}_\ell) \\ &= \frac{f_{\mathbf{R}_\ell|V_\ell}(\mathbf{r}_\ell|g)}{\sum_{h\in\mathbb{F}_q} f_{\mathbf{R}_\ell|V_\ell}(\mathbf{r}_\ell|h)} = \frac{e^{-\frac{\|\mathbf{r}_\ell - \mathbf{e}_g\|^2}{2\tau^2}}}{\sum_{h\in\mathbb{F}_q} e^{-\frac{\|\mathbf{r}_\ell - \mathbf{e}_h\|^2}{2\tau^2}}} \\ &= \frac{e^{\frac{\mathbf{r}_\ell(g)}{\tau^2}}}{\sum_{h\in\mathbb{F}_q} e^{\frac{\mathbf{r}_\ell(h)}{\tau^2}}}. \end{aligned} \tag{8}$$

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

$$\mathbb{E}[\mathbf{S}_\ell|\mathbf{R}_\ell = \mathbf{r}_\ell] = \sum_{g\in\mathbb{F}_q} \mathbf{e}_g \Pr(\mathbf{S}_\ell = \mathbf{e}_g|\mathbf{R}_\ell = \mathbf{r}_\ell). \tag{9}$$

A variant of this approach can be found in [31] for a system without an outer code. Ideally, we would like to take advantage of the outer code with the more precise MMSE estimate of the form $\mathbb{E}[\mathbf{S}_\ell|\mathbf{R} = \mathbf{r}]$. Unfortunately, as mentioned above, computing this conditional expectation is far too complex to be implemented in practice. A viable alternative that trades off performance and complexity is to execute BP on the factor graph of the outer LDPC code. Implicitly, this approach computes an estimate for every $\mathbf{S}_\ell$ based on the computation tree of the code, up to a certain depth [8], [40].

Frameworks to perform BP on factor graphs are well-established [41]; thus, we assume some familiarity with such iterative procedures. We proceed by first considering the nuances of non-binary LDPC factor graphs, then presenting a BP algorithm, and finally proposing a dynamic denoiser for SR-LDPC codes.

A. Non-Binary LDPC Graphs

The factor graph for an $\mathbb{F}_q$ LDPC code features L variable (left) nodes and $L(1-R)$ check (right) nodes enforcing parity constraints, where R is the *design rate* of the LDPC code [6]–[8]. An important distinction between binary and non-binary LDPC codes is that a factor graph for the latter code typically includes edge labels. These labels take values in $\mathbb{F}_q \setminus \{0\}$. A vector $\mathbf{v} \in \mathbb{F}_q^L$ is a valid codeword if it satisfies the parity equations

$$\sum_{v_\ell \in N(c_p)} \omega_{\ell,p} \otimes v_\ell = 0 \qquad \forall p \in [L(1-R)], \tag{10}$$

where $N(c_p)$ is the collection of variable nodes adjacent to parity check c_p. The summation and the multiplication operator $\otimes$ in (10) 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 [41], [42], 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 (10) can be expressed as an indicator function

$$\mathcal{G}_p(\mathbf{v}_p) = \mathbf{1}\left(\sum_{v_\ell \in N(c_p)} \omega_{\ell,p} \otimes v_\ell = 0\right) \tag{11}$$

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

$$\mathcal{G}(\mathbf{v}) = \prod_{p\in[L(1-R)]} \mathcal{G}_p(\mathbf{v}_p). \tag{12}$$

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

Authorized licensed use limited to: Texas A M University. Downloaded on September 30,2024 at 11:53:43 UTC from IEEE Xplore. Restrictions apply.

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 by $\boldsymbol{\mu}_{v\to c}$, and messages in the reverse direction are represented by $\boldsymbol{\mu}_{c\to v}$. Formally, a message going from check node c_p to variable node $v_\ell \in N(c_p)$ is computed component-wise through

$$\boldsymbol{\mu}_{c_p\to v_\ell}(g) = \sum_{\mathbf{v}_p : v_\ell = g} \mathcal{G}_p(\mathbf{v}_p) \prod_{v_j \in N(c_p)\setminus v_\ell} \boldsymbol{\mu}_{v_j\to c_p}(g_j). \quad (13)$$

While (13) 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 (10) or, equivalently,

$$\sum_{v_j \in N(c_p)\setminus v_\ell} \omega_{j,p} \otimes g_j = -\omega_{\ell,p} \otimes g. \quad (14)$$

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

$$\boldsymbol{\mu}_{v_\ell\to c_p}(g) \propto \boldsymbol{\alpha}_\ell(g) \prod_{c_\xi \in N(v_\ell)\setminus c_p} \boldsymbol{\mu}_{c_\xi\to v_\ell}(g). \quad (15)$$

The '$\propto$' symbol indicates that the positive measure can be normalized before being sent out as a message. Vector $\boldsymbol{\alpha}_\ell$ in (15) can be viewed as a collection of beliefs based on local observations, as in (8). That is, entry $\boldsymbol{\alpha}_\ell(g)$ captures the probability that symbol g is the true field element within section ℓ. Other BP messages are initialized with $\boldsymbol{\mu}_{v\to c} = \mathbf{1}$ and $\boldsymbol{\mu}_{c\to v} = \mathbf{1}$. The traditional parallel sum-product algorithm iterates between (13) and (15), 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 established literature, we define two actions.

Definition 2 (Vector $+g$ Operator [7])**.** *For field element $g \in \mathbb{F}_q$, the vector $+g$ operator acting on $\mathbf{b}$ 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 3 (Vector $\times g$ Operator [7])**.** *For field element $g \in \mathbb{F}_g \setminus \{0\}$, we define the vector $\times g$ operator acting on $\mathbf{b}$ 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 $\mathbb{F}_q$.

Under these operations, we can rewrite (13) in a concise manner:

$$\boldsymbol{\mu}_{c_p\to v_\ell} = \left(\bigodot_{v_j \in N(c_p)\setminus v_\ell} \left(\boldsymbol{\mu}_{v_j\to c_p}\right)^{\times \omega_{j,p}^{-1}} \right)^{\times(-\omega_{\ell,p})} \quad (16)$$

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

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

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

$$\overline{\boldsymbol{\mu}}_{v_j\to c_p} = \left(\boldsymbol{\mu}_{v_j\to c_p}\right)^{\times \omega_{j,p}^{-1}} \quad (17)$$

$$\overline{\boldsymbol{\mu}}_{c_p\to v_\ell} = \left(\boldsymbol{\mu}_{c_p\to v_\ell}\right)^{\times\left(-\omega_{\ell,p}^{-1}\right)}. \quad (18)$$

Then (16) morphs into the simpler expression

$$\overline{\boldsymbol{\mu}}_{c_p\to v_\ell} = \bigodot_{v_j \in N(c_p)\setminus v_\ell} \overline{\boldsymbol{\mu}}_{v_j\to c_p}. \quad (19)$$

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 (15) also admits a more compact form. For $p \in N(v_\ell)$, the traditional outgoing message from a variable node can be written as

$$\boldsymbol{\mu}_{v_\ell\to c_p} = \frac{\boldsymbol{\alpha}_\ell \circ \left(\bigcirc_{c_\xi \in N(v_\ell)\setminus c_p} \boldsymbol{\mu}_{c_\xi\to v_\ell}\right)}{\left\|\boldsymbol{\alpha}_\ell \circ \left(\bigcirc_{c_\xi \in N(v_\ell)\setminus c_p} \boldsymbol{\mu}_{c_\xi\to v_\ell}\right)\right\|_1} \quad (20)$$

where $\circ$ represents the Hadamard product.

The process of passing messages from variable nodes to their neighboring check nodes and then from check nodes to their neighboring variable nodes is referred to as one BP iteration. After performing one or more BP iterations, a natural estimate for the distribution associated with variable node v_ℓ, including intrinsic information, is obtained as

$$\hat{\mathbf{S}}_\ell = \frac{\boldsymbol{\alpha}_\ell \circ \left(\bigcirc_{c_p \in N(v_\ell)} \boldsymbol{\mu}_{c_p\to v_\ell}\right)}{\left\|\boldsymbol{\alpha}_\ell \circ \left(\bigcirc_{c_p \in N(v_\ell)} \boldsymbol{\mu}_{c_p\to v_\ell}\right)\right\|_1}. \quad (21)$$

As we will shortly see, (21) is the output of our proposed denoiser.

Remark 4. *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* (19)*, with*

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

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.

Authorized licensed use limited to: Texas A M University. Downloaded on September 30,2024 at 11:53:43 UTC from IEEE Xplore. Restrictions apply.

C. Sparse Regression LDPC Denoiser

Conceptually, one can initiate the state of the LDPC factor graph using the effective observation $\mathbf{r}$, run BP, and then gain an estimate for the state based on (21). As mentioned before, in the absence of BP iterations, local estimates reduce to the conditional expectation $\mathbb{E}[\mathbf{S}_\ell|\mathbf{R}_\ell = \mathbf{r}_\ell]$ found in (9). Yet, as more iterations of the BP algorithm are performed, the estimate for $\mathbf{S}_\ell$ is refined based on the computation tree of the outer code, up to a certain depth.

Definition 5 (BP-N Denoiser)**.** *Let N denote the number of BP iterations to perform, where N is less than the girth of the LDPC factor graph. The BP-N denoiser*

1) *initializes the LDPC factor graph with estimates $\boldsymbol{\alpha}_\ell$ computed from $\mathbf{r}_\ell$ for $\ell \in [L]$ according to* (8)*;*
2) *computes and passes variable to check messages (see* (20)*) and check to variable messages (see* (17)*,* (19)*,* (18) *and Remark 4) along the edges of the factor graph in an alternating fashion N times;*
3) *computes updated state estimates according to* (21)*.*

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.

The derivation of the Onsager correction term corresponding to this denoiser is included in [39], Appendix A. At this point, we turn to a performance assessment of sparse regression LDPC codes via numerical simulations.

IV. Simulation Results

In this section, we seek to characterize the performance of a randomly generated SR-LDPC code[1]. As benchmarks, we compare our results to the concatenated SPARC/LDPC scheme presented in [32] as well as an iterative coding scheme involving 5G-NR binary LDPC codes and 4-PAM. The gap between the unconstrained capacity and the capacity constrained to 4-PAM inputs is small at the chosen rate; this makes the latter comparison meaningful.

The scenario of interest is one in which 5888 information bits are to be transmitted over 7350 real channel uses. We define the SR-LDPC rate 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 ($R_{\text{LDPC}} \approx 0.96$) over GF(256) whose edges are generated via progressive edge growth (PEG) and whose weights were chosen uniformly at random from the elements of $\mathbb{F}_{256} \setminus 0$. During the decoding process, AMP is run 25 times and, within AMP iteration i, $i+1$ BP iterations are run, where the iterations are indexed using zero-based indexing. After the AMP-BP iterations have finished, 100 additional BP iterations are run. If at any point during the decoding process a valid codeword is obtained, the decoding process is terminated.

The alternate iterative scheme proceeds as follows. The information bits are encoded using a 5G-NR LDPC code of rate $R \approx 0.4$ and then mapped to 4-PAM symbols so the total number of channel uses is fixed at 7350. At the receiver, a soft-input-soft-output (SISO) demapper produces LLRs which are fed to a SISO LDPC decoder. The extrinsic information provided by the SISO LDPC decoder is fed back to the demapper as priors on the transmitted bits and the demapper produces a new set of LLRs. This process is repeated multiple times, after which hard bit decisions are made.

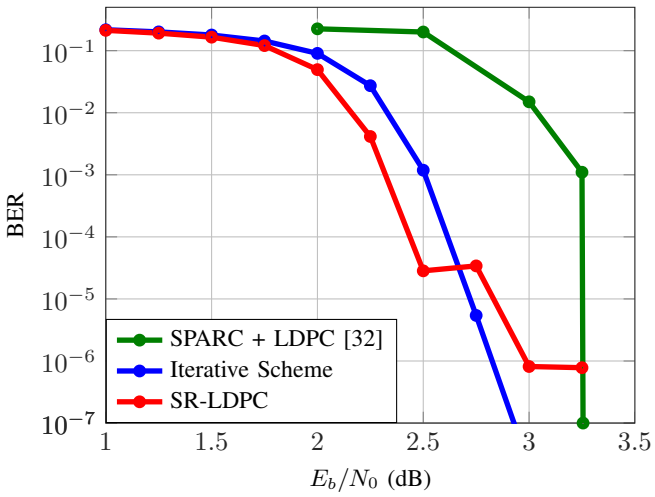

Fig. 1. BER performance comparison of a SR-LDPC code, the scheme from [32], and an interative LDPC + 4-PAM scheme.

A BER comparison of these three schemes is included in Fig. 1. It is clear that this SR-LDPC code provides a nearly 1 dB improvement at a BER of 10^{-3} over the other SPARC/LDPC concatenated coding structure. Furthermore, the SR-LDPC code outperforms the iterative scheme for moderate BERs; however, the selected SR-LDPC scheme does have a higher error floor than both the iterative scheme and the scheme from [32]. This may warrant further investigation.

V. Conclusion

In conclusion, this article presents sparse regression LDPC codes and their decoding. A sparse regression code is a concatenated structure with an inner SPARC-like code and an outer non-binary LDPC code. The inner code is decoded using AMP with a dynamic denoiser which runs BP on the factor graph of the outer LDPC code to improve the state estimate at every iteration. After running many iterations of the AMP-BP decoder, the final effective observation is fed into a standard LDPC BP decoder, which seeks to correct any residual errors that may be present in the received signal. Numerical results show that sparse regression codes exhibit performance improvements over standard LDPC codes and other concatenated SPARC/LDPC constructions. Directions for future work include optimizing the outer LDPC code and applying coded demixing [38] to SR-LDPC codes to accommodate multiple users.

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

Authorized licensed use limited to: Texas A M University. Downloaded on September 30,2024 at 11:53:43 UTC from IEEE Xplore. Restrictions apply.

,(((,QWHUQDWLRQDO6PSRVLPRQ,QIRUPDWLRQ7KHRU,6,7

REFERENCES

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

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

[3] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 585–598, 2001.

[4] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, 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. Inform. Theory*, vol. 47, no. 2, pp. 657–670, 2001.

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

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

[8] T. J. Richardson and R. L. Urbanke, *Modern coding theory*, Cambridge University 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 Communications Magazine*, vol. 52, no. 7, pp. 168–176, 2014.

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

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

[12] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Spatially coupled ensembles universally achieve capacity under belief propagation," *IEEE Trans. Inform. Theory*, vol. 59, no. 12, pp. 7761–7813, 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. Inform. Theory*, vol. 60, no. 11, pp. 6943–6965, 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. Inform. Theory*, vol. 60, no. 12, pp. 7389–7415, 2014.

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

[16] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary ldpc codes over gf (q)," *IEEE Trans. on Commun.*, vol. 55, no. 4, pp. 633–643, 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. on Commun.*, vol. 58, no. 5, pp. 1365–1375, 2010.

[18] B.-Y. Chang, D. Divsalar, and L. Dolecek, "Non-binary protograph-based LDPC codes for short block-lengths," in *Information Theory Workshop*. IEEE, 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. Inform. Theory*, vol. 58, no. 5, pp. 2541–2557, 2012.

[20] A. Joseph and A. R. Barron, "Fast sparse superposition codes have near exponential error probability for $R < C$," *IEEE Trans. Inform. Theory*, vol. 60, no. 2, pp. 919–942, 2013.

[21] R. Venkataramanan, S. Tatikonda, and A. Barron, "Sparse regression codes," *Foundations and Trends in Communications and Information Theory*, vol. 15, no. 1–2, pp. 1–195, 2019.

[22] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[23] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489–509, 2006.

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

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

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

[27] M. Bayati, M. Lelarge, and A. Montanari, "Universality in polytope phase transitions and message passing algorithms," *The Annals of Applied Probability*, vol. 25, no. 2, pp. 753–822, 2015.

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

[29] J. Barbier and F. Krzakala, "Replica analysis and approximate message passing decoder for superposition codes," in *Proc. Int. Symp. on Information Theory*. IEEE, 2014, pp. 1494–1498.

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

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

[32] A. Greig and R. Venkataramanan, "Techniques for improving the finite length performance of sparse superposition codes," *IEEE Trans. on Commun.*, vol. 66, no. 3, pp. 905–917, 2017.

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

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

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

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

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

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

[39] J. R. Ebert, J.-F. Chamberland, and K. R. Narayanan, "On sparse regression ldpc codes," *arXiv preprint arXiv:2301.01899*, 2023.

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

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

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

[43] H. Song and J. R. Cruz, "Reduced-complexity decoding of Q-ary LDPC codes for magnetic recording," *IEEE Trans. on Magnetics*, vol. 39, no. 2, pp. 1081–1087, 2003.

[44] A. Goupil, M. Colas, G. Gelle, and D. Declercq, "FFT-based BP decoding of general LDPC codes over Abelian groups," *IEEE Trans. on Commun.*, vol. 55, no. 4, pp. 644–649, 2007.

Authorized licensed use limited to: Texas A M University. Downloaded on September 30,2024 at 11:53:43 UTC from IEEE Xplore. Restrictions apply.