Received 19 April 2020; accepted 3 May 2020. Date of publication 12 May 2020; date of current version 3 June 2020.

Digital Object Identifier 10.1109/OJCOMS.2020.2994048

Decoding Rate-Compatible 5G-LDPC Codes With Coarse Quantization Using the Information Bottleneck Method

MAXIMILIAN STARK [1] (Student Member, IEEE), LINFANG WANG [2] (Student Member, IEEE), GERHARD BAUCH [1] (Fellow, IEEE), AND RICHARD D. WESEL [2] (Fellow, IEEE)

[1] Institute of Communications, Hamburg University of Technology, 21073 Hamburg, Germany

[2] Department of Electrical and Computer Engineering, University of California at Los Angeles, Los Angeles, CA 90095, USA

CORRESPONDING AUTHOR: M. STARK (e-mail: maximilian.stark@tuhh.de)

This work was supported in part by the Open Access Funds of the Hamburg University of Technology (TUHH) in the funding programme Open Access Publishing, in part by the National Science Foundation under Grant CCF-1911166, in part by Physical Optics Corporation, and in part by SA Photonics.

ABSTRACT Increased data rates and very low-latency requirements place strict constraints on the computational complexity of channel decoders in the new 5G communications standard. Practical low-density parity-check (LDPC) decoder implementations use message-passing decoding with finite precision, which becomes coarse as complexity is more severely constrained. In turn, performance degrades as the precision becomes more coarse. Recently, the information bottleneck (IB) method was used to design mutual-information-maximizing mappings that replace conventional finite-precision node computations. As a result, the exchanged messages in the IB approach can be represented with a very small number of bits. 5G LDPC codes have the so-called protograph-based raptor-like (PBRL) structure which offers inherent rate-compatibility and excellent performance. This paper extends the IB principle to the flexible class of PBRL LDPC codes as standardized in 5G. The extensions include IB decoder design for puncturing and rate-compatibility. In contrast to existing IB decoder design techniques, the proposed decoder can be used for a large range of code rates with a static set of optimized mappings. The proposed construction approach is evaluated for a typical range of code rates and bit resolutions ranging from 3 bit to 5 bit. Frame error rate simulations show that the proposed scheme always outperforms min-sum decoding algorithms and operates close to double-precision sum-product belief propagation decoding. Furthermore, alternatives to the lookup table implementations of the mutual-information-maximizing mappings are investigated.

INDEX TERMS LDPC codes, 5G, message-passing decoding, mutual-information based signal processing, information bottleneck method, machine learning.

I. INTRODUCTION

LOW-DENSITY parity-check (LDPC) codes are used in the current 5G standard due to their very powerful error-correction performance [1]. However, to fully exploit the error-correction capabilities of these codes, high precision belief propagation is required. In practice, this decoding approach comes with two main challenges. First, the messages which carry the soft information are exchanged iteratively between the check nodes and variable nodes in the Tanner graph and require a high resolution to precisely convey the belief on a codeword bit. As a result, the message transfer becomes a major bottleneck as the block length increases [2]. Second, realizing the correct check node operation, i.e., the box-plus operation requires several computationally complex operations. The traditional strategies to combat the high message resolution and the computational burden at the check nodes involve quantization of the messages and approximation of the arithmetical operations [3]. However, it is well known that the performance deteriorates drastically as the quantization of the messages is too coarse, i.e., below 6 bits.

One way of dealing with coarse quantization is the finite-alphabet iterative decoding (FAID) approach [4], [5]. Here, hand-optimized lookup tables are designed which replace the conventional node operations. The FAID approach was shown to achieve very competitive performance on a binary symmetric channel with regular LDPC codes, despite coarse quantization.

However, in [6], [7], a fundamentally different way to design node operations tailored to coarse quantization was sketched. Instead of trying to approximate the arithmetic in the node operations, a mutual-information-based design was proposed. Here, the operations do not mimic the actual box-plus operation but represent a discrete input-output relation which maximizes the mutual information between the quantized message and the corresponding codeword bit. In [8] it was shown that the ideas from [6] and [7] can be directly linked to the *information bottleneck method* [9], which is a more generic framework with roots in machine learning and information theory [10].

While similar in operation to the lookup tables developed for the FAID approach [4], [5], the tables used in information bottleneck (IB) decoders are designed analytically [11]–[15]. Maximizing mutual information requires access to the respective joint distributions in each decoder iteration. These distributions can be tracked and predicted using discrete density evolution [7], [11], [13], [14]. In turn, no log-likelihood ratios (LLRs) are processed in the entire decoder at any time. Instead, integer-valued messages, referred to as *cluster indices* in this paper, are exchanged. The resulting IB LDPC decoders operate only 0.1dB away from the double-precision belief propagation performance, even though all messages were represented with 4 bits and the operations were simple discrete input-output mappings [11], [13], [14]. In [16], it was shown that with similar decoders, a decoding throughput up to 588 Gb/s is possible with high energy and area efficiency.

Interestingly, despite several successful applications of mutual-information-based signal processing, for some time, designing LDPC decoders for non-optimized irregular LDPC codes remained an open problem. Results in [14], [17] suggested that both, coarsely quantized min-sum decoders and decoders leveraging mutual-information-maximizing lookup tables only work for certain irregular LDPC codes with optimized degree distributions. However, in [12], 4 bit information bottleneck decoders that use a technique called message alignment were presented. These decoders approach the performance of double-precision belief-propagation decoders also for irregular LDPC codes without specifically optimized degree distributions.

To the best of our knowledge, all information bottleneck decoders and related mutual-information-maximizing lookup table decoders in literature are tailored for a particular code ensemble with a specific rate and do not take into account puncturing. However, in practical systems a rate-compatible decoding scheme is favorable. Recently, so-called protographbased raptor-like (PBRL) LDPC codes were shown to pair very powerful error-correcting capabilities and an efficient structure that enables an inherent rate-compatibility [18]. This family of LDPC codes is also used in the 5G standard [1]. In particular, PBRL LDPC codes leverage degree-one variable nodes which allow for a flexible rate change. These degree-one variable nodes build an incremental-redundancy code in addition to the high-rate mother code. As a result, variable nodes with degree one up to twenty might exist in the Tanner graph. This high irregularity in addition to puncturing requires a very evolved IB design approach to build decoders that work with a very coarse quantization. In addition, the proposed decoder is not matched to a particular code rate, as state-of-the-art IB decoders are [11], [13], [14]. The decoder, defined by a single set of lookup tables, can be used for the entire range of rates covered by the PBRL LDPC code. In the conference version of this paper [19], the IB decoder design only for puncturing was sketched. This paper extends this concept to rate-compatible design with efficient table-reuse across various code rates. Here, message alignment is leveraged as a general design concept. As message alignment turns out as a fundamental technique crucial to build powerful IB decoders, we propose a novel view on message alignment which allows achieving even better performance in terms of frame error rates compared to the results presented in [19]. As a further extension to [19], this journal paper investigates alternative implementations of the mutual-information-maximizing lookup tables. Here, a static min-sum-inspired mapping at the check node is investigated which still outperforms the conventional min-sum decoder.

In detail, the paper contains the following main contributions:

- Extension of the design of IB LDPC decoders from [11], [12] to include puncturing in both the high-rate mother code and the degree-one variable nodes of PBRL codes.
- Reformulation of message alignment as an IB problem, facilitating designs for irregular LDPC codes.
- The new interpretation of message alignment allows the reuse of tables across the entire rate range allowing a compact rate-compatible IB decoder for an entire PBRL code family.
- Investigation of several message alignment implementations and their effect on the decoder performances.
- A 4-bit information bottleneck decoder for a PBRL code family designed using the new construction approach that outperforms a 6-bit normalized-min-sum decoder and performs very close to double-precision belief propagation decoding.
- Detailed investigation of the impact of the bit resolution on the decoding performance. Therefore, different resolutions, from 3 bit to 5 bit are considered. It is shown that even 3 bit quantization is still sufficient to outperform a 4 bit min-sum decoder.
- Investigation of the effect of alternative implementations of the mutual-information-maximizing lookup tables.

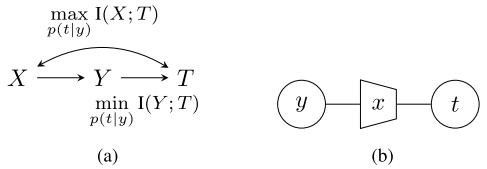


FIGURE 1. (a) Illustration of the information bottleneck setup, (b) Exemplary information bottleneck graph.

Organisation: The IB method and PBRL LDPC codes are briefly reviewed in Section II. In Section III, we summarize the design of IB LDPC decoders and discuss differences to conventional decoding techniques like belief-propagation decoding or min-sum decoding. Thereafter, *message alignment* and its variants are reviewed. Section V targets the effect of puncturing in PBRL LDPC codes and techniques to incorporate puncturing in the decoder design. Finally, this paper targets the problem of rate-compatible decoding architectures in Section VI. In Section VII, further numerical simulations comparing the performance of our proposed decoder with several reference systems and LDPC codes are provided. Additionally, simulation results for different implementations of the mutual-information-preserving mappings and message alignment are provided in Section VII. Section VIII concludes the paper.

Notation: The realizations $y \in \mathcal{Y}$ from the event space $\mathcal{Y}$ of a discrete random variable Y occur with probability $\Pr(Y = y)$ and $p(y)$ is the corresponding probability distribution. The cardinality or alphabet size of a random variable is denoted by $|\mathcal{Y}|$. Joint distributions and conditional distributions are denoted $p(x, y)$ and $p(x|y)$, respectively.

II. PREREQUISITES AND PRIOR ART

This section briefly reviews the information bottleneck method and its applications in signal processing. Furthermore binary PBRL LDPC codes are introduced.

A. THE INFORMATION BOTTLENECK METHOD

Formally, the information bottleneck method is a clustering framework. It pairs ideas from machine learning, i.e., decision theory, with information-theoretical concepts, i.e., mutual information and the Kullback-Leibler divergence. In contrast to rate-distortion theory, which focuses on compression with respect to a given distortion measure, the information bottleneck setup involves the pairwise mutual information between three random variables. This setup is sketched in Figure 1(a). The random variable X is termed the *relevant* random variable, Y denotes the *observed* random variable and T denotes the *compressed* random variable. Furthermore, these three random variables form the Markov chain $X \to Y \to T$. Typically, in mutual-information-based signal processing, the focus is solely on the maximum preservation of relevant information given a certain cardinality $|\mathcal{T}|$. In [20] it was shown that in this case the, in general, probabilistic clustering $p(t|y)$ becomes a deterministic relation $t = f(y)$, which maps an observation $y \in \mathcal{Y}$ into a cluster $t \in \mathcal{T}$. For a more detailed review of information bottleneck algorithms, we refer to [20]. In summary, the general objective in mutual-information-based signal processing is to obtain a mapping $t = f(y)$ such that

$$\max_{t=f(y)} \mathrm{I}(X; T) \tag{1}$$

with an inherent constraint on $|\mathcal{T}|$ as $t \in \mathcal{T}$. As a by-product, an information bottleneck algorithm delivers the *meaning* of each cluster, i.e., $p(x|t)$.

B. MUTUAL-INFORMATION-MAXIMIZING LOOKUP TABLES

Due to the discrete nature of the event space $\mathcal{T}$ of T the function $t = f(y)$ maps a continuous or discrete input y onto a discrete output t. In literature, different strategies to implement this function exist. Proposed approaches range from threshold-based quantizers to lookup tables which store the input-output relation $t = f(y)$ by storing the respective t for every y. Figure 1(b) uses the information bottleneck graph notation introduced in [8] to express this relation. The input y of the shown lookup table is compressed by the mutual-information-maximizing mapping such that the output t is highly informative about the relevant variable X.

Please note that at no point the function or lookup table is intended to approximate or simplify an arithmetic function. In other words, mutual-information-based signal processing does *not* start from a given arithmetic expression and does *not* try to find a smart approximation. Instead, it takes quantization effects and message resolutions into account right from scratch. The rest of this paper uses the terms $f(y)$, lookup table and the mapping $p(t|y)$ from Figure 1(a) synonymously.

C. PROTOGRAPH-BASED RAPTOR-LIKE (PBRL) LDPC CODES

Thorpe [21], [22] introduced LDPC codes constructed from a protograph, which is a small Tanner graph that describes the connectivity of the overall LDPC Tanner graph. A copy and permute operation referred to as "lifting" obtains the full LDPC parity check matrix from the protograph.

Figure 2 shows the protograph structure of a PBRL code as described in [18], [23]. The protograph of a PBRL LDPC code consists of two parts: (1) a highest-rate code (HRC) protograph and (2) an incremental redundancy code (IRC) protograph. The IRC provides lower rates as more of its variable nodes are transmitted, starting from the top. For a more detailed introduction to PBRL LDPC codes we refer the reader to [18], [23]. In general, the protomatrix of the protograph shown in Figure 2 is given as

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{HRC} & \mathbf{0} \\ \mathbf{H}_{IRC} & \mathbf{I} \end{bmatrix}, \tag{2}$$

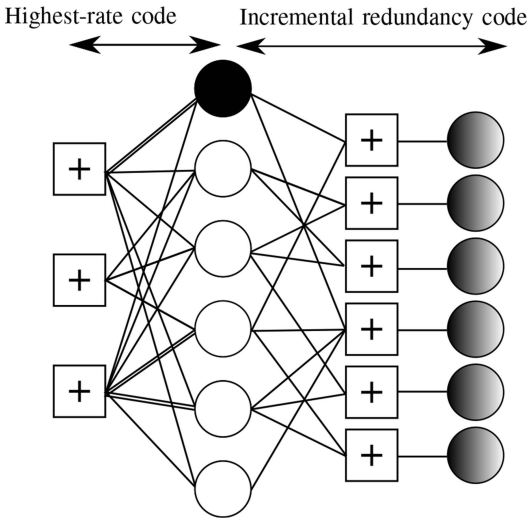


FIGURE 2. Protograph of a PBRL LDPC code where the shaded node depicts a puntured variable node in the highest-rate code and the partial shade indicates that degree-one variable nodes can be punctured to adapt the rate.

where $\mathbf{H}_{HRC}$ denotes the parity check matrix of the highest rate code, $\mathbf{H}_{IRC}$ denotes the parity check matrix of the incremental redundancy code and $\mathbf{0}$ and $\mathbf{I}$ denote the all-zeros and the identity matrix respectively.

This paper addresses the issue of designing IB decoders that accommodate the puncturing that is inherent to PBRL code families. As a design principle of PBRL codes, one or two variable nodes in the HRC remain punctured for all supported code rates [23], as indicated by the shaded HRC variable node in Figure 2. Thus, the IB decoder for the HRC must be designed to handle this puncturing. Additionally, all of the IRC variable nodes are punctured for the highest code rate, but degree-one variable nodes are added to the protograph as the rate is lowered. Thus, a degree-one variable node might be punctured depending on the code rate, as indicated by the partial shade of the degree-one variable nodes. The IB decoder must be able to adapt to the induced changes in the degree distributions and the associated changes in the probability distributions of message reliabilities that occur as the rate is lowered.

III. INFORMATION BOTTLENECK DECODERS FOR UNPUNCTURED BINARY LDPC CODES

In recent works [11]–[14], information bottleneck decoders were shown to handle the trade-off between low implementation complexity and near-optimal performance very well. In the following section, this paper reviews all required steps to construct such information bottleneck decoders for unpunctured binary LDPC codes.

A. TRANSMISSION SCHEME AND CHANNEL OUTPUT QUANTIZATION

We consider a binary LDPC encoded transmission over a quantized output, symmetric additive white Gaussian noise (AWGN) channel with binary phase-shift keying modulation (BPSK). We denote the equally likely transmitted symbols

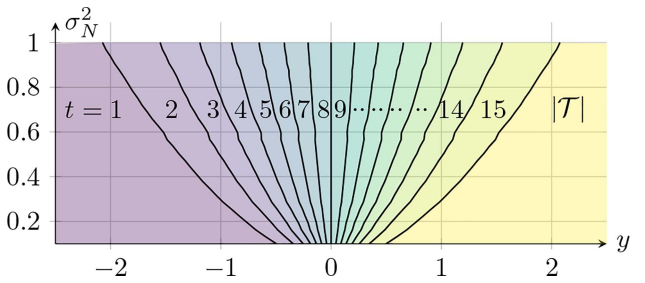


FIGURE 3. Quantization boundaries and regions for the BI-AWGN channel computed using the information bottleneck algorithm from [11] for different noise variances σ_N^2.

by x, which serve as channel input. The binary channel input and continuous channel output y are related by the transition probability $p(y|x)$. Feeding $p(y, x)$ and the cardinality $|\mathcal{T}_{ch}|$ into the information bottleneck algorithm yields the quantizer mapping $p(t_{\text{ch}}|y)$, where $t_{\text{ch}} \in \mathcal{T}_{ch}$ denotes the discrete channel output. Such a mapping is sketched in Figure 3 where the lines illustrate the boundaries of the clusters t_{ch} for different noise variances σ_N^2. In general, a representative log-likelihood ratio (LLR) can be assigned to each quantization region. These representative LLRs correspond to the quantized channel knowledge which serves as input for belief-propagation decoding. In contrast, an information bottleneck decoder does not use any quantized LLRs, but processes only the abstract quantization index $t_{\text{ch}} \in \{1, \ldots, |\mathcal{T}_{ch}|\}$ instead.

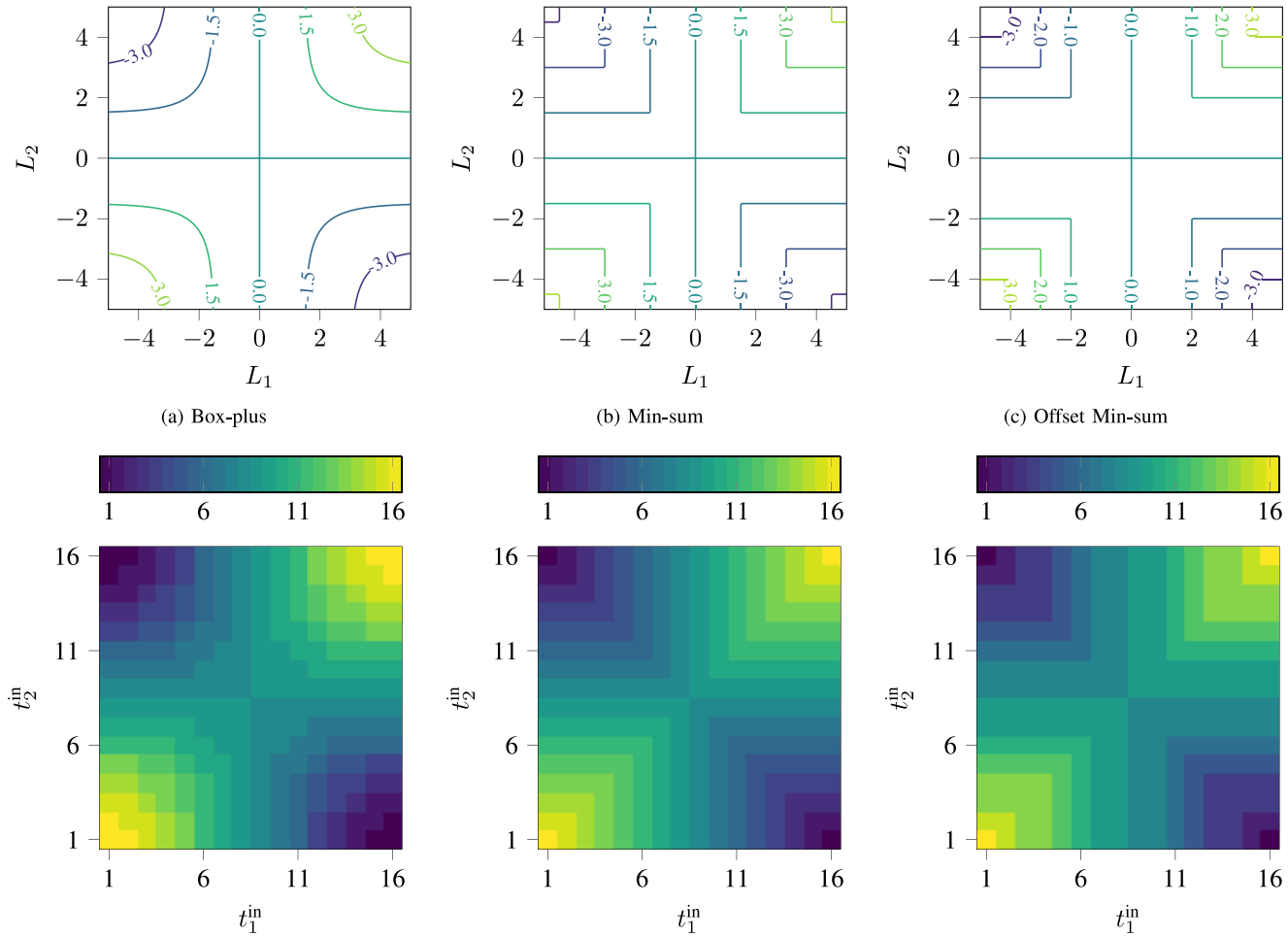
B. INFORMATION BOTTLENECK DECODERS FOR REGULAR LDPC CODES

1) CONVENTIONAL VARIABLE NODE OPERATION

In state-of-the-art belief propagation decoding, the soft-information is represented by LLRs. At a variable node, all incoming LLRs are summed up except the message received over the edge for which extrinsic information is to be generated.

2) MUTUAL-INFORMATION-BASED VARIABLE NODE OPERATION

In mutual-information-based signal processing the task is to determine a deterministic function which maps an input vector $\mathbf{t}^{\text{in}} = [t_1^{\text{in}}, \ldots, t_M^{\text{in}}]^{\mathsf{T}}$ with M incoming discrete, integer valued messages t_i^{in}, $i = 1, \ldots, M$ into a cluster t^{out}. In this case, the relevant variable X is the codeword bit represented by the variable node. The clustering is done such that the mutual information between the compressed observation t^{out} and the relevant codeword bit is maximized. Consequently, the actual decoding simplifies to an exchange of cluster indices and discrete mappings or look-up operations. Thus, there is no need to exchange real-valued LLRs and to perform arithmetic operations. Instead, the challenge is to obtain $p(x, \mathbf{t}^{\text{in}})$ such that the information bottleneck algorithm can find the optimal assignment $p(t^{\text{out}}|\mathbf{t}^{\text{in}})$ or $t^{\text{out}} = f(\mathbf{t}^{\text{in}})$. A detailed derivation of $p(x, \mathbf{t}^{\text{in}})$ is beyond the scope of this paper but can be found in [7], [8], [14].



(a) Box-plus (b) Min-sum (c) Offset Min-sum

(d) Mutual-information-maximizing look-up table (e) Cluster-based min-sum (f) Cluster-based offset min-sum

FIGURE 4. **Input-output relation of (a) the box-plus operation, (b) min-sum operation, (c) offset min-sum operation, (d) the check node lookup table designed using the information bottleneck method, (e) the min-sum operation using cluster indices and (f) the offset min-sum operation using cluster indices.**

3) CONVENTIONAL CHECK NODE OPERATION

As benchmarks, this paper considers three conventional implementations of the check node operation. Their input-output relations are visualized in Figure 4(a) - Figure 4(c). The axes display the possible input values and the color or contour displays the respective output value. In state-of-the-art belief propagation decoding evaluating the check node equation equals the box-plus sum of the incoming log-likelihood ratios. The box-plus operation $\boxplus$ of two LLRs L_1 and L_2 is defined as

$$L_1 \boxplus L_2 = \log \frac{e^{L_1} e^{L_2} + 1}{e^{L_1} + e^{L_2}}.$$

To avoid the evaluation of the exponential and logarithmic functions, a common approximation is the min-sum approximation. Here it is assumed that

$$L_1 \boxplus L_2 \approx \text{sign}(L_1) \cdot \text{sign}(L_2) \cdot \min\{|L_1|, |L_2|\}.$$

The input-output relation of the box-plus operation is sketched in Figure 4(a) and respectively for the min-sum operation in Figure 4(b). Clearly, the non-linearities of the box-plus operation result in bent contour plots, whereas the min-sum operation cannot capture these curves and produces an edged shape instead.

In [3] two versions of the min-sum operation were proposed. First, the normalized min-sum decoder weights the minimum LLR by a factor α which yields

$$L_1 \boxplus L_2 \approx \text{sign}(L_1) \cdot \text{sign}(L_2) \cdot \frac{\min\{|L_1|, |L_2|\}}{\alpha}. \quad (3)$$

Due to the scaling of the LLRs by $1/\alpha$ the performance can be largely improved compared to pure min-sum decoding [3]. The proper choice of $1/\alpha$ can be determined using density evolution.

Second, instead of multiplying with a constant $1/\alpha$ the offset-min-sum decoder subtracts a predetermined constant β depending on the smallest magnitude of the respective LLRs, i.e.,

$$L_1 \boxplus L_2 \approx \text{sign}(L_1) \cdot \text{sign}(L_2) \times \max((\min\{|L_1|, |L_2|\} - \beta), 0). \quad (4)$$

As pointed out in [3], in contrast to normalized min-sum the constant β will set LLRs with small magnitudes to zero, i.e., the contribution in the next variable node update vanishes. In Figure 4(c) the input-output relation for an offset-min-sum check node is sketched. In Figure 4(c), the values of β were set to

$$\beta = \begin{cases} 0, & \text{for } \min\{|L_1|, |L_2|\} < 1 \\ 1, & \text{for } 1 \leq \min\{|L_1|, |L_2|\} < 6 \\ 2, & \text{for } 6 \leq \min\{|L_1|, |L_2|\}. \end{cases} \tag{5}$$

4) MUTUAL-INFORMATION-BASED CHECK NODE OPERATION

To determine the relevant-information-preserving mapping, the joint distribution $p(x, \mathbf{t}^{\text{in}})$ of the input vector $\mathbf{t}^{\text{in}}$ pooling discrete, integer valued messages and the relevant quantity is required. For generation of extrinsic information at a check node, the relevant variable X is the modulo 2 sum of $d_c - 1$ bits connected to the check node. Applying the information bottleneck algorithm yields a discrete input-output mapping $t^{\text{out}} = f(\mathbf{t}^{\text{in}})$ as depicted in Figure 4(d) for $\mathbf{t}^{\text{in}} = [t_1^{\text{in}}, t_2^{\text{in}}]^{\mathsf{T}}$. Here, the clusters t_1^{in} and t_2^{in} are sorted according to there respective LLR. Although not intended to approximate the box plus operation, the mapping found by the information bottleneck represents the bended contours of the box plus operations much better than the min-sum operation. Furthermore, one observes that the symmetric properties of the box plus operations are preserved which allows to reduce the memory need when storing the function $t^{\text{out}} = f(\mathbf{t}^{\text{in}})$ as look-up table.

Interestingly, as pointed out in [24] when the variable node operations are replaced by mutual-information-maximizing look-up tables an application of the min-sum approximation is straightforward if the incoming messages are discrete cluster indices t_1^{in} and t_2^{in} and not LLRs L_1 and L_2. This is possible if the natural ordering of the cluster indices t_i^{in} represents the ordering of the LLRs L_i associated with t_i^{in}, where $i \in \{1, \ldots, M\}$ and M denotes the number of processed messages. The respective input-output relations for min-sum and offset min-sum using clusters are shown in Figure 4(e) and Figure 4(f) respectively.

C. RELEVANT-INFORMATION-PRESERVING CLUSTERINGS FOR ARBITRARY IRREGULAR LDPC CODES

In contrast to regular LDPC codes, irregular LDPC codes are characterized by nodes with varying degrees, i.e., the number of incoming messages differs. This paper leverages the edge-degree distribution [25]:

$$\lambda(\zeta) = \sum_{d=2}^{\lambda_{\max}} \lambda_d \zeta^{d-1} \quad \rho(\zeta) = \sum_{d=2}^{\rho_{\max}} \rho_d \zeta^{d-1}, \tag{6}$$

where λ_d denotes the fraction of edges connected to variable nodes with degree d and ρ_d denotes the fraction of edges connected to check nodes with degree d. Thus, for irregular LDPC codes the input joint distribution $p(x, \mathbf{t}^{\text{in}}|d)$ for the information bottleneck depends on the node degree d. Consequently, it is not sufficient to design message mappings only for variable nodes or check nodes but for variable nodes or check nodes *considering* the individual node degrees.

In density evolution a code ensemble is considered, i.e., instead of a particular irregular LDPC code with a certain parity check matrix, the connectivity between variable and check nodes is only known on average defined by the degree distribution. To construct the required input joint distributions $p(x, \mathbf{t}^{\text{in}}|d)$, discrete density evolution from [7] needs to be extended to consider the degree distribution of the code ensemble. In order to incorporate the degree distribution one has to average over all possible degrees resulting in the marginal distribution $p(x, \mathbf{t}^{\text{in}})$, i.e.,

$$p\left(x, \mathbf{t}^{\text{in}}\right) = \sum_{d=2}^{\lambda_{\max}} \lambda_d p\left(x, \mathbf{t}^{\text{in}}|d\right). \tag{7}$$

In *discrete* density evolution $p(x, t^{\text{out}})$ has to be tracked instead of $p(x, \mathbf{t}^{\text{in}})$. We define the marginal distribution $p(x, t^{\text{out}})$ as

$$\begin{aligned} p\left(x, t^{\text{out}}\right) &= \sum_{d=2}^{\lambda_{\max}} \lambda_d p\left(x, t^{\text{out}}|d\right) \\ &= \sum_{d=2}^{\lambda_{\max}} \lambda_d \sum_{\mathbf{t}^{\text{in}} \in \mathcal{T}^{\text{vec}}} p\left(t^{\text{out}}|\mathbf{t}^{\text{in}}, d\right) p\left(x, \mathbf{t}^{\text{in}}|d\right), \end{aligned} \tag{8}$$

where $\mathcal{T}^{\text{vec}}$ denotes the set of all possible combinations of $\mathbf{t}^{\text{in}}$ for a node with degree d. As it will be shown later, this straightforward marginalization is unfavorable for the mutual-information-maximizing design principle.

In [17], it was first described that discretized min-sum decoders require a particular degree distribution $\lambda(\zeta)$, respectively $\rho(\zeta)$, to not suffer from a large gap between the decoding threshold of the belief-propagation decoder and the decoding threshold of the discretized min-sum decoder. Note that the eventspace $\mathcal{T} = \{1, \ldots, |\mathcal{T}|\}$ is *independent* of the node degree d but in contrast the particular meaning $p(x|t^{\text{out}}, d)$ *depends* on the node degree. Thus, from the cluster indices alone, the check node cannot resolve if a message originates from a variable with high or low degree. As the variety of node degrees increases, the dynamic range of the respective reliabilities also increases. Thus, especially irregular LDPC codes with high irregularity suffer from large performance degradation when decoded with coarse quantization and conventional LLR-based decoding, e.g., using min-sum decoding. However, in the next section, this paper discusses a pre-processing step to improve the performance of the information bottleneck decoder independent of the degree distribution.

IV. MESSAGE ALIGNMENT AS AN INFORMATION BOTTLENECK PROBLEM

This section reviews message alignment as introduced in [26]. In addition to the approach from [26], an alternative realization is proposed which is closely related to [14].

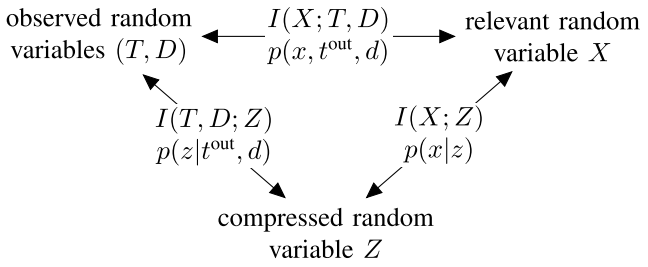


FIGURE 5. Message alignment formulated as an information bottleneck, where $I(X;Z)$ is the relevant information, $I(X;T,D)$ is the original mutual information and $I(Y;T,D)$ is the compressed information.

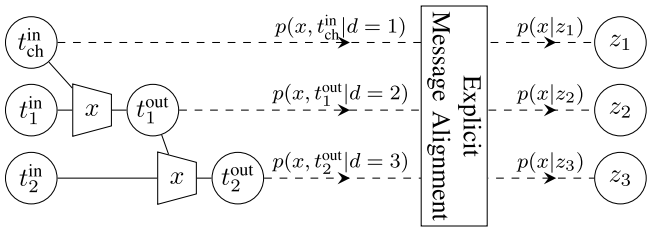


FIGURE 6. Designing explicit message alignment: The joint distribution $p(x, t^{\text{out}}, d)$ used for alignment is composed of the individual output distributions $p(x, t^{\text{out}}|d)$ weighted by the edge-degree distribution.

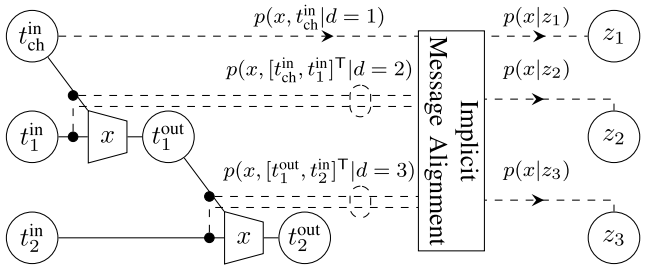


FIGURE 7. Designing implicit message alignment: The joint distribution $p(x, \mathbf{t}^{\text{in}}, d)$ used for alignment is now composed of the individual input distributions $p(x, \mathbf{t}^{\text{in}}|d)$ of the information bottleneck algorithm weighted by the edge-degree distribution.

Therefore, message alignment is posed as an information bottleneck problem which makes it more intuitive to generalize message alignment for puncturing as it is proposed in Section V.

A. EXPLICIT MESSAGE ALIGNMENT

As the event space $\mathcal{T}$ is *independent* of the node degree d but in contrast, $p(x|t^{\text{out}}, d)$ *depends* on the node degree, the marginalization as in (8) averages misaligned beliefs [12]. These beliefs do not represent the density evolution equations appropriately. Instead of performing (8) directly, first, the problem needs to be considered from an information theoretical perspective, e.g., using the information bottleneck framework.

For the node-dependent IB design, the information bottleneck setting involves the random variables T, D, X and Z as depicted in Figure 5. As visualized in Figure 6, given $p(x, t^{\text{out}}|d)$ and $p(d)$ given by λ_d in the considered example, the joint distribution of these random variables can be found as $p(x, t, d)$, with mutual information $I(X;T,D)$. As the outgoing message shall be restricted to $\mathcal{Z}$, the task is to find

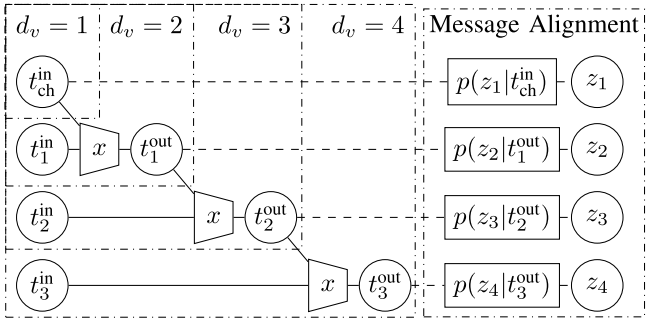


FIGURE 8. Processing in a concatenated lookup table for $d_V = 4$ with message alignment.

a mapping $p(z|t^{\text{out}}, d)$ such that $I(X;Z)$ is maximized. This paper assumes $\mathcal{Z} = \mathcal{T}$. As the mapping $p(z|t^{\text{out}}, d)$ can be decomposed and embedded in the node design, such that the lookup table becomes $p(z|\mathbf{t}^{\text{in}}, d)$ instead of $p(t^{\text{out}}|\mathbf{t}^{\text{in}}, d)$, this technique is called *message alignment* as it ensures that messages with the same index capture the same belief. Since the alignment relies explicitly on the degree-dependent mappings $p(t^{\text{out}}|\mathbf{t}^{\text{in}}, d)$ it is referred to as *explicit* message alignment.

Example 1: Let us assume an IB variable node with degree $d_v = 4$, depicted in Figure 8 as concatenation of two-input-lookup tables. In Figure 8 each lookup table is depicted as trapezoid with the input vector $\mathbf{t}^{\text{in}} = [t_{\text{ch}}^{\text{in}}, t_1^{\text{in}}]^{\mathsf{T}}$ or $\mathbf{t}^{\text{in}} = [t_{i-1}^{\text{out}}, t_i^{\text{in}}]^{\mathsf{T}}$, where $i = 2, \ldots, d_v - 1$ and output t_i^{out}. As illustrated in Figure 8, the lookup tables for all node degrees $d_v < 4$ are implicitly constructed as they serve as intermediate results for the degree $d_v = 4$ variable node. Thus, the overall number of lookup tables depends only on the largest node degree and not on the variety of node degrees. In turn, the intermediate mappings $p(t^{\text{out}}|\mathbf{t}^{\text{in}}, d)$ are fed into the message alignment unit to explicitly construct a node-degree-independent belief $p(x, z)$.

B. IMPLICIT MESSAGE ALIGNMENT

As an extension to the original message alignment approach, this paper discusses an alternative approach similar to [14]. Instead of treating message alignment as post-processing step, it can also be included as a design objective immediately in the look-up table design. Here, the mappings $p(z|\mathbf{t}^{\text{in}})$ are designed using $p(x, \mathbf{t}^{\text{in}}|d)$ directly instead of using $p(x, t^{\text{out}}|d)$. This is depicted in Figure 7. Thus, analog to the message alignment setup from Figure 5, now the random variables $\mathbf{T}, X, D$ and Z serve as a starting point. Thus, instead of two subsequent optimizations, i.e., first to find $p(t^{\text{out}}|\mathbf{t}^{\text{in}}, d)$ and then $p(z|t^{\text{out}}, d)$, as in the explicit message alignment setting, $p(z|\mathbf{t}^{\text{in}}, d)$ is found in one shot. The results obtained with the two techniques are discussed and compared later in this paper.

V. INFORMATION BOTTLENECK DECODERS FOR PBRL LDPC CODES

To decode PBRL LDPC codes, the respective IB decoders must support puncturing. Puncturing denotes the process of *not* transmitting code bits. As a result, the number of

information bits per code word bits, which is the code rate, can be easily and gradually changed. At the receiver side in a conventional LLR-based decoder, the punctured bits are represented by an LLR zero which is fed into the decoder. Thus, although puncturing itself is a fairly easy problem for conventional decoders, it is not straightforward for information bottleneck decoders. This section describes one of the main contributions of this paper, partially proposed in the conference version of this paper in [19]. First, incorporating punctured nodes using message alignment is shown. Then, in addition to [19], this section focuses on the different notions of puncturing faced in PBRL codes and provides detailed examples.

A. CONSTRUCTING INFORMATION BOTTLENECK DECODERS FOR PUNCTURED PBRL LDPC CODES

As it was shown in [7], [11], to achieve the best performance with mutual-information-based lookup tables, symmetric input distributions are optimum if the channel is symmetric. As a result, the LLR zero is originally not covered in IB decoders. To tackle this problem, an additional cluster might be introduced which explicitly corresponds to the LLR zero. This approach results in an uneven number of clusters.

This paper shows, that by using message alignment and the structure of PBRL codes, mutual-information maximizing lookup tables that support puncturing can be designed also with an even number of clusters that show close-to-optimum decoding performance. First, the effects of puncturing at the variable nodes and check nodes are investigated with respect to the computation of the joint distributions.

1) PUNCTURING FROM A VARIABLE NODE PERSPECTIVE

A variable node can face puncturing in two ways. First, the channel message which is connected to the variable node can be punctured. Second, a message from a check node can be punctured. Irrespective of the origin, a punctured message cannot contribute any *relevant* information. As a result, there is no need to process this message.

Example 2: Let us assume again an IB variable node with degree $d_v = 4$, depicted in Figure 8 as a concatenation of two-input-lookup tables. Please remember that in the unpunctured case the number of messages processed was $M = 4$, since three messages received over edges connected to check nodes plus the channel message are processed. If the channel message is punctured, the effective degree is reduced by one. Thus, the respective input distribution is $p(x, t_1^{\text{in}}, t_2^{\text{in}}, t_3^{\text{in}})$. Also, if the message from a check node is punctured, the effective degree is reduced by one. Thus, the respective input distribution is for example $p(x, t_{\text{ch}}^{\text{in}}, t_1^{\text{in}}, t_2^{\text{in}})$ if t_3^{in} is punctured. Please note that for this joint distribution it does not matter which of the messages conveyed by check nodes is punctured, as due to density evolution and message alignment all individual distributions $p(x, t_i^{\text{in}})$ are the same in each iteration. In turn, only the number of punctured messages conveyed from the check nodes matters.

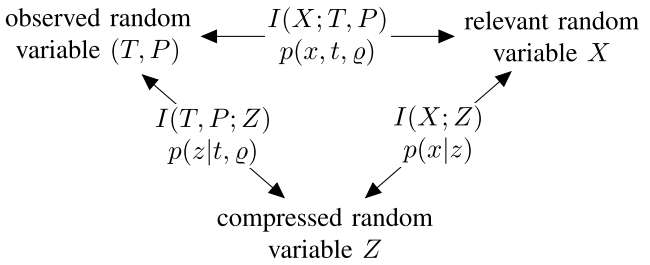


FIGURE 9. Considering puncturing as message alignment problem, where $I(X;Z)$ is the relevant information, $I(X;T,P)$ is the original mutual information and $I(Z;T,P)$ is the compressed information.

This example illustrates that two notions of puncturing exist at a variable node. We will refer to the first type as channel-induced puncturing and the second type as check-node-induced puncturing. First, we introduce the random variable P with event space $\varrho \in \{\text{true}, \text{false}\}$ indicating if a node is punctured or not. In this paper, the puncturing rate equals the fraction of variable nodes with degree $d > 1$ that are punctured.

CHANNEL-INDUCED PUNCTURING

As depicted in Figure 8 the channel message is processed in the first stage, for which the input vector is $\mathbf{t}^{\text{in}} = [t_{ch}^{in}, t_1^{in}]^{\mathsf{T}}$. The resulting joint distribution equals

$$p\left(x, \left[t_{\text{ch}}^{\text{in}}, t_1^{\text{in}}\right]^{\mathsf{T}}\right) = \frac{1}{p(x)} p\left(x, t_{\text{ch}}^{\text{in}}\right) p\left(x, t_1^{\text{in}}\right). \tag{9}$$

Clearly, $p(x, [t_{\text{ch}}^{\text{in}}, t_1^{\text{in}}]^{\mathsf{T}})$ and thus also $p(x|t_1^{\text{out}})$, which is used in the next step, depends on the statistics of the quantized channel output (see Section III-A). When incorporating puncturing, $p(x, t_{ch}^{in})$ differs if the channel message is punctured or not. As a result, we rewrite (9) as

$$p\left(x, \left[t_{\text{ch}}^{\text{in}}, t_1^{\text{in}}\right]^{\mathsf{T}} | \varrho\right) = \frac{1}{p(x)} p\left(x, t_{\text{ch}}^{\text{in}} | \varrho\right) p\left(x, t_1^{\text{in}}\right). \tag{10}$$

Due to the concatenation of lookup tables as shown in Figure 8 all subsequent tables depend on P. Consequently, in a straightforward implementation, the number of required lookup tables will increase drastically to account for all possible combinations of punctured and non-punctured nodes and their respective degrees. Hence, this paper proposes to make use of the message alignment technique to prohibit such an increase in the number of look-up tables. The corresponding setting is shown in Figure 9. By applying message alignment, one creates the mapping $p(z|t, \varrho)$ and the meaning $p(x|z)$ such that all subsequently constructed tables do not depend any longer on the node being punctured or not. This approach ensures that the number of lookup tables is not increased as compared to an unpunctured IB decoder.

CHECK-NODE-INDUCED PUNCTURING

Besides the puncturing of the channel message, also a message received from a check node can be punctured, e.g., if the respective check node is connected to a punctured

degree-one variable node. This is explained in more detail in Section V-A2. As a result, the variable node degree is reduced, i.e., fewer lookup tables need to be constructed (see Figure 8). However, the computation of the joint distributions remains

$$p\left(x, \left[t_{\text{ch}}^{\text{in}}, t_1^{\text{in}}\right]^{\mathsf{T}}\right) = \frac{1}{p(x)} p\left(x, t_{\text{ch}}^{\text{in}}\right) p\left(x, t_1^{\text{in}}\right). \tag{11}$$

for the first lookup table and

$$p\left(x, \left[t_{i-1}^{\text{out}}, t_i^{\text{in}}\right]^{\mathsf{T}}\right) = \frac{1}{p(x)} p\left(x, t_{i-1}^{\text{out}}\right) p\left(x, t_i^{\text{in}}\right) \tag{12}$$

for all subsequent lookup tables $i = 2, \ldots, d_{v,\max} - 1$. However the overall *effective* edge-degree distribution λ_{eff} will be changed.

2) PUNCTURING FROM A CHECK NODE PERSPECTIVE

Check nodes are only implicitly affected by puncturing if they are connected to a punctured degree-one variable node, or in the first iteration if the incoming message is a punctured channel message. If one incoming message is punctured, i.e., the relevant information is zero, all outgoing messages will also be punctured, i.e., they convey no information. Thus, the respective check node is effectively *deactivated* in the Tanner graph. This changes the effective edge-degree distribution for the check nodes ρ_{eff} which has to be considered in the message alignment for the lookup table construction.

3) COMPUTING THE EFFECTIVE DEGREE DISTRIBUTIONS

As discussed in the previous section, puncturing effects the effective degrees of both the variable nodes and the check nodes. Thus, in contrast to classical density evolution where the code ensemble is considered, when designing information bottleneck decoders the Tanner graph needs to be known to determine the effective edges in PBRL codes and the corresponding effective degree distributions $\rho_{\text{eff}} \neq \rho$ and $\lambda_{\text{eff}} \neq \lambda$.

VI. CONSTRUCTING RATE-COMPATIBLE INFORMATION BOTTLENECK DECODERS

Rate-compatible codes which allow to efficiently adapt the code rate according to the channel conditions are a crucial and inevitable part of modern communication systems. In contrast to state-of-the-art message-passing decoders, their mutual-information-based counterparts are not rate-compatible as the set of mutual information preserving mappings is matched to a specific rate. This section contains one of our main contributions. In this section, we devise an approach to reuse the mappings across several rates.

A. REUSING TABLES OF INFORMATION BOTTLENECK DECODERS FOR MULTIPLE RATES

As summarized in Section II, PBRL codes consist of a high-rate code (HRC) and an incremental redundancy code (IRC). The HRC can be described by the triplet of parameters (n_v, n_c, n_p), where n_v is the number of variable nodes,

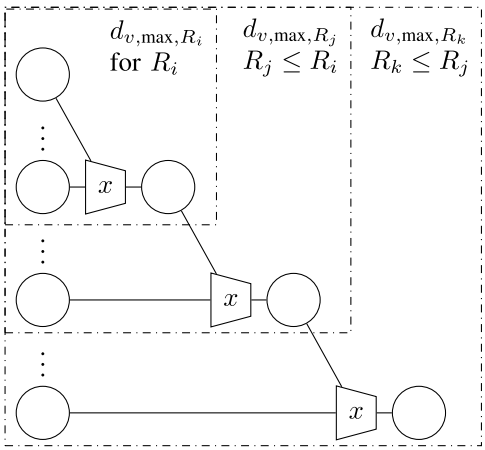


FIGURE 10. Schematic sketch of reuse of lookup tables for all rates based on the lookup tables for codes with higher rates.

n_c denotes the number of check nodes and n_p indicates the number of punctured nodes in the HRC. The set of possible rates R_i of a PBRL code with full rank $\mathbf{H}_{HRC}$ is given by

$$R_i = \frac{(n_v - n_c)}{(n_v - n_p + i)} \tag{13}$$

where $i \leq 0$ indicates the number of unpunctured degree-one variable nodes and let $i_{\max}$ denote the maximum number of degree-one nodes added by the IRC. As discussed in the previous section, puncturing degree-one variable nodes has an impact on the effective degree-distributions λ_{eff}, ρ_{eff} and, thus, also the maximum node degree $\lambda_{\text{eff},\max}$ depends on the number of punctured degree-one variable nodes. At the lowest rate, no degree-one variable node is punctured. Thus, one will observe the largest values $\lambda_{\text{eff},\max}$ across all rates R_i for $R_{i_{\max}}$. On the other hand, $\lambda_{\text{eff},\max}$ will be smallest for R_0

Proposition 1: For a fixed PBRL code with full rank $\mathbf{H}_{HRC}$, $\lambda_{\text{eff},\max,R_i} \leq \lambda_{\text{eff},\max,R_j}, \forall i, j = 0, \ldots, i_{\max}$ if $R_i > R_j$.

Proof: As the IRC adds more redundancy by activating parts in the Tanner graph, this is equivalent to augmenting $\mathbf{H}_{HRC}$. Hence, the node degree of the variable nodes can only be increased and not decreased. □

Please note, that the number of needed lookup tables depends on the node degree (see Figure 8). According to Proposition 1, designing an information bottleneck decoder for the highest code rate supported by a PBRL code yields variable nodes with the smallest number of lookup tables. Thus, similar to the table reuse in an irregular LDPC code, where a node with larger degree is obtained by stacking new tables on top of a node with lower degree, we propose to use the lookup tables for the highest rate as a starting point for the design of the lookup tables for lower rates. This is depicted in Figure 10.

Example 3: Let us consider a fixed PBRL code with rates $R_i = 2/3, R_j = 1/2, R_k = 1/3$ and $d_{v,\max,R_i} = 9$, $d_{v,\max,R_j} = 15$, $d_{v,\max,R_k} = 27$. Please note, that $d_{c,\max} = 19$

TABLE 1. Simulation parameters.

decoder	node operation (check / var)	precision exchanged messages	precision check node	precision variable node	channel quantizer
belief-propagation	box-plus / addition	64 bit	64 bit	64 bit	None
offset min-sum	(4) / addition	4 bit	4 bit	6 bit	4 bit
NMSA	(3) / addition	6 bit	6 bit	6 bit	None
proposed	lookup table / lookup table	4 bit	4 bit	4 bit	4 bit

and is independent of the chosen rate. Without reuse, mutual-information-maximizing mappings are designed for each rate for a fixed design-E_b/N_0. Results for such a setting are shown later in Section VII-A. With the reuse, first the mappings for rate $R_i = 2/3$ with $d_{v,\max,R_i} = 9$ are designed for a fixed design-E_b/N_0 optimized for this rate. In the second step, the mappings derived for $R_i = 2/3$ are reused for $R_j = 1/2$ with $d_{v,\max,R_j} = 15$. As $d_{v,\max,R_i} - 1$ mappings could be reused, only $d_{v,\max,R_j} - d_{v,\max,R_i} = 6$ new mappings are designed and appended as shown in Figure 10. These new mappings are designed for a new design-E_b/N_0 optimized for $R_j = 1/2$. Thus, a subset of mappings is used mismatched, i.e., designed for another rate and also different channel conditions. However, it will be shown in Section VII-D that only a small performance degradation will be observed. This is due to the fact that message alignment is adapted to the degree distribution and compensates the slight imperfections of the reused mappings. In the next step, the mappings for R_k are designed with a new design-E_b/N_0 optimized for $R_k = 1/3$ but reusing the $d_{v,\max,R_i} - 1$ mappings for $R_i = 2/3$ and the $d_{v,\max,R_j} - d_{v,\max,R_i}$ mappings optimized for $R_j = 1/2$. Please note that message alignment is not shown in Figure 10 explicitly but it is done for every code rate successively. In addition, also the mappings in the check nodes remain unchanged and only message alignment is updated if needed.

VII. RESULTS AND DISCUSSION

In this section, we present and discuss results obtained performing frame error rate simulations for an exemplary PBRL LDPC code. We propose to construct all involved lookup tables just once for a fixed design-E_b/N_0 which is optimized for each rate. The constructed lookup tables are then stored and applied for all E_b/N_0. Hence, the lookup table construction needs to be done only once and offline. In Section VII-A, the proposed decoder which incorporates puncturing is evaluated. Afterwards, in Section VII-B, the impact of the bit resolution on the performance is analyzed. Section VII-C discusses the impact of explicit and implicit message alignment (see Section IV) on the frame error rate performance. Simulation results for the proposed table reuse strategy from Section VI are shown in Section VII-D. Finally, in Section VII-E, alternative implementations of the lookup table approach as proposed in [24] are applied to the proposed design approach.

We consider three reference schemes to compare the performance of our decoder. The decoding of a codeword is stopped after a maximum number of 100 decoding iterations or earlier if the syndrome check is successful. First, we consider a double-precision belief propagation decoder with a flooding schedule. The received samples are *not* coarsely quantized but represented with double precision and the internal operations are additions at the variable node and box-plus at the check node. Second, we use the normalized min-sum algorithm (NMSA) [27], [28] with 6 bit resolution for the outgoing check node message and 6 bit for the outgoing variable node message. Again the inputs to the decoder are *not* coarsely quantized but represented with double precision. The operations here are additions at the variable nodes but the normalized min-sum approximation is used at the check nodes (see (3)). Third, we use the offset-min-sum decoder with only 4 bit resolution at the check node and offsets according to (5) and 6 bit at the variable node to prevent an overflow when adding the 4 bit messages received from the channel quantizer. Finally, we designed our proposed information bottleneck decoder for fully 4 bit integer architecture. This means, starting from the channel quantizer which outputs 4 bit integers, the internal messages require only 4 bit and only *lookup* operations are performed. These lookups do not mimic any arithmetic function but realize the relevant-information preserving mappings found using the information bottleneck method.

As discussed in the introduction, the FAID approach from [5] is conceptually related to the decoders from [11] or [14]. However, to the best of our knowledge, all these decoder do not support puncturing or rate-compatibility and are thus not shown as benchmark systems.

A. INCORPORATING PUNCTURING USING MESSAGE ALIGNMENT

In this subsection, we investigate the proposed generalized decoder design to cover punctured variable nodes. Here, the code was taken from [23]. The code has $K = 1032$ information bits and is evaluated for various code rates R_c ranging from $R_c = 1/3$ up to $R_c = 4/5$. Furthermore, in this subsection, the decoder mappings were designed for each code rate individually with an individual design-E_b/N_0, i.e., the table reuse from Section VI is not applied.

The most important parameters of the applied decoders are summarized in Table 1 for a quick overview. First, we consider a decoder designed for a fixed rate of $R_c = 0.5$. The results are shown in Figure 11. As expected, the belief-propagation (BP) algorithm (●-marker) achieves the best frame error rate performance, but at the same time, has the highest computational complexity (see Table 1). Although all applied operations in the information bottleneck decoder (⧫-marker) are simple lookups, the decoder performs only less than 0.2 dB worse than the benchmark. The results are even more remarkable when considering the tremendous gap to the

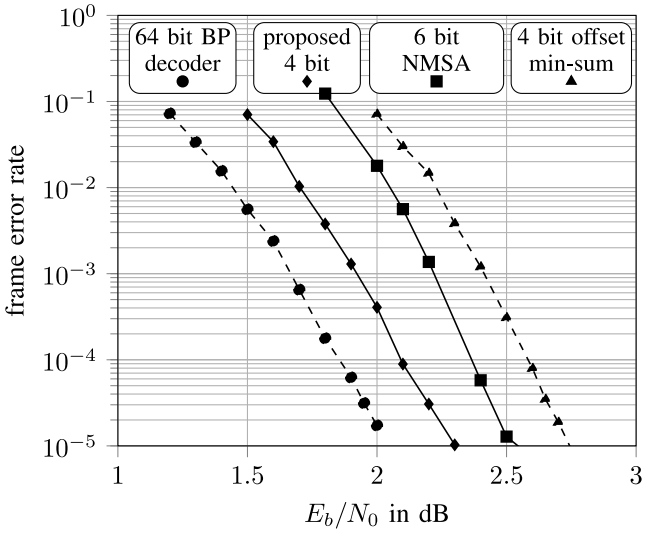


FIGURE 11. Frame error rates for the proposed scheme (diamond-marker), and the reference schemes summarized in Table 1 for the considered PBRL LDPC code with code rate $R_C = 1/2$.

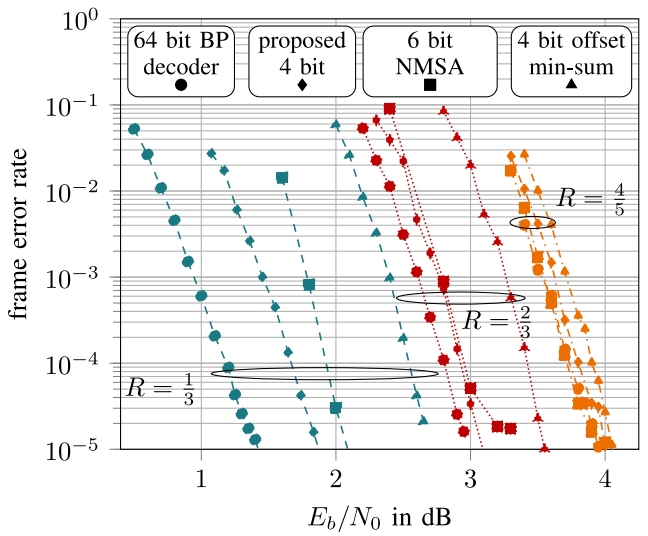


FIGURE 12. Frame error rates for the proposed scheme (diamond-marker) and the reference schemes summarized in Table 1 for the considered PBRL LDPC code with code rate $R_C = 1/3$ (blue, dashed), 2/3 (dark red, dotted), 4/5 (dark orange, dash dot).

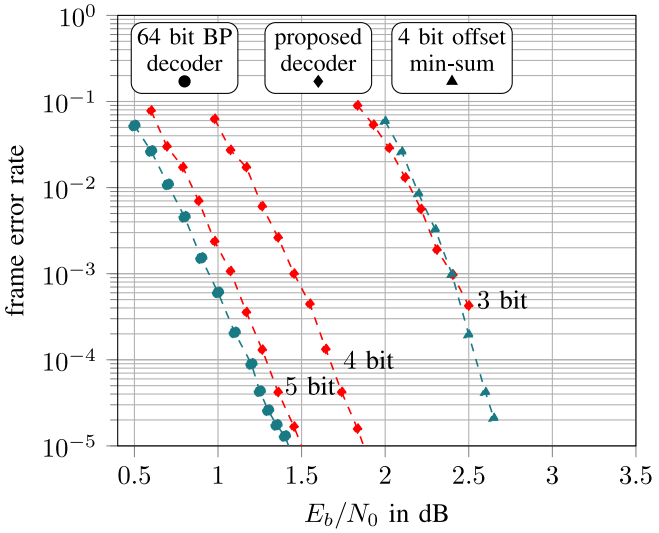


FIGURE 13. Frame error rate simulations for the proposed decoder for different bit resolution as summarized in Table 2 for code rate $R_C = 1/3$. Only belief propagation and the offset min-sum decoder are shown as reference, with parameters according to Table 1.

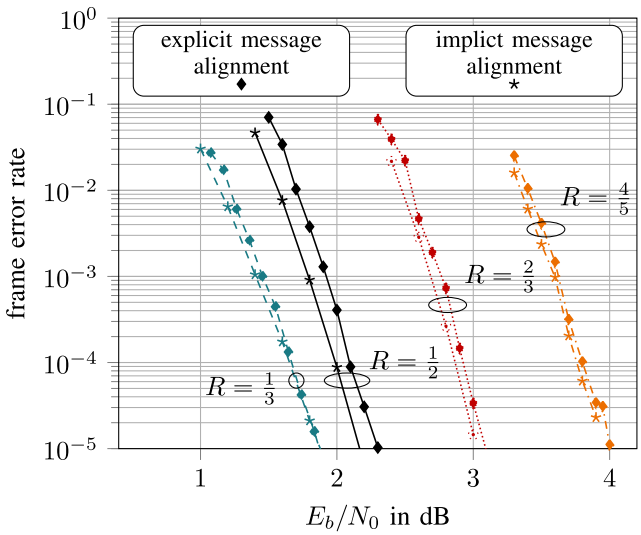
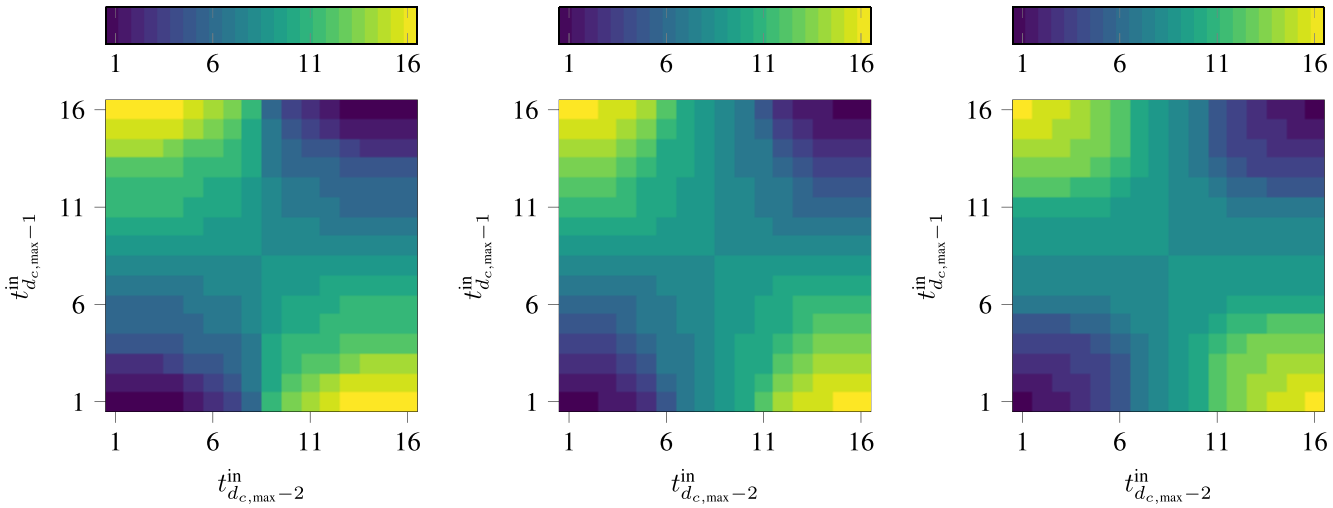


FIGURE 14. Frame error rate simulations for implicit and explicit message alignment.

offset-min-sum and normalized min-sum decoders with an even slightly higher resolution. Please note, that PBRL codes have typically variable nodes with very large degrees. From the gap of 0.75 dB noticed in Figure 11, we conclude that a conventional offset-min-sum decoder, which exchanges only 4 bit messages, cannot be used for PBRL codes with such a coarse quantization since the dynamic range of the LLRs cannot be captured appropriately. The gap can be reduced by choosing a finer resolution as indicated by the frame error rate curve for the 6 bit NMSA decoder. However, with the generalized design for information bottleneck decoders proposed in this paper, both challenges, i.e., puncturing and rate-compatible design, can be efficiently tackled to enable fully 4 bit decoders for PBRL codes. Figure 12 shows the results for various other rates. For all considered rates, the belief propagation decoder with double-precision resolution and no channel quantizer achieves the best performance. However, again we observe that the proposed information bottleneck decoder operates very close to this benchmark. Interestingly, the proposed schemes outperform the 4 bit offset min-sum decoder and the 6 bit NMSA decoder for all investigated rates.

B. IMPACT OF THE BIT RESOLUTION ON THE DECODER PERFORMANCE

For the considered code, this subsection investigates the impact of the chosen bit resolution on the performance. The respective bit resolutions used are summarized in Table 2. The results are shown in Figure 13. For the sake of clarity,



(a) Last check node mapping at iteration 1 (b) Last check node mapping at iteration 50 (c) Last check node mapping at iteration 100

FIGURE 15. Input-output relation of the check node lookup table designed using the information bottleneck method at different iterations.

TABLE 2. Simulation parameters of investigated information bottleneck decoders.

exchanged messages	check node	variable node	channel quantizer
3 bit	3 bit	3 bit	3 bit
4 bit	4 bit	4 bit	4 bit
5 bit	5 bit	5 bit	5 bit

TABLE 3. Memory requirements per iteration for information bottleneck decoders with and without table reuse for a 4bit decoder.

R_c	$d_{c,\max}$	check node memory		$d_{v,\max}$	variable node memory	
		no reuse	reuse		no reuse	reuse
4/5	19	8.7kB	8.7kB	6	2.6kB	2.6kB
2/3	19	8.7kB	-	9	4.1kB	1.5kB
1/2	19	8.7kB	-	15	7.2kB	3.1kB
1/3	19	8.7kB	-	27	13.3kB	6.1kB
Total		34.8kB	8.7kB		27.1kB	13.3 kB

only the results for $R_c = 1/3$ are shown and the reference systems are limited to the offset min-sum decoder and the belief propagation decoder with the parameters from Table 1. However, similar results were obtained for all other code rates. Interestingly, it can be observed that for a 5 bit information bottleneck decoder, the performance gap to double-precision belief propagation decoding nearly vanishes. Furthermore, it can be observed that the proposed 3 bit information bottleneck decoder shows the same performance as the offset min-sum decoder, which uses 4 bit for the channel quantizer, 4 bit for the exchanged messages and 6 bit for the variable node operation (see Table 1).

C. IMPACT OF DIFFERENT IMPLEMENTATIONS OF MESSAGE ALIGNMENT FOR PROPOSED IB DECODERS

As proposed in Section IV, the message alignment approach can be realized either based on $p(x, t^{\text{out}}|d)$ termed explicit message alignment or based on $p(x, \mathbf{t}^{\text{in}}|d)$ referred to as implicit message alignment. In Figure 14 the impact of the selected message alignment approach on the frame error rate performance is investigated. It is shown, that the performance gain achieved by the implicit approach is 0-0.1 dB over the explicit message alignment approach. The slight performance degradation of explicit message alignment is caused by using the compressed representation t^{out} of $\mathbf{t}^{\text{in}}$ in the alignment step instead of $\mathbf{t}^{\text{in}}$. However, when considering the concatenated scheme with reuse, the implicit message alignment approach has slightly higher memory complexity. The implicit alignment mapping has $|\mathcal{T}|^2$ input combinations whereas explicit message alignment works on the compressed random variable directly and has only $|\mathcal{T}|$ input combinations (see Figure 6 and Figure 7).

D. MEMORY CONSIDERATIONS AND TABLE REUSE

Besides supporting puncturing, the proposed generalized decoder design enables also the reuse of lookup tables across several rates. In contrast to state-of-the-art information bottleneck decoders where one set of tables was designed for only one particular rate, the proposed decoder using the technique proposed in Section VI uses one set of tables for all rates. In Figure 16, the simulation results are shown, where the proposed information bottleneck decoder optimized for each rate from the previous section is included as a reference. As described in Section VI, the decoder construction starts with the highest code rate, i.e., $R_c = 4/5$. Thus, no difference between the decoders can be observed for this rate. The lookup tables for all lower code rates are built on top of the lookup tables from the code with a higher rate. Here, a small performance degradation below 0.1 dB can be observed due to the mismatched table reuse.

According to [12], the memory of one two-input lookup table is given as $\frac{|\mathcal{T}|^2 \cdot |\mathcal{T}|}{8}$ byte if $|\mathcal{T}_{ch}| = |\mathcal{T}|$ and $d_c - 2$ tables are needed for a check node with degree d_c and $d_v - 1$ tables for a variable node with degree d_v. Table 3 summarizes the overall required memory demand. It can be observed that for the considered PBRL code, the memory per iteration

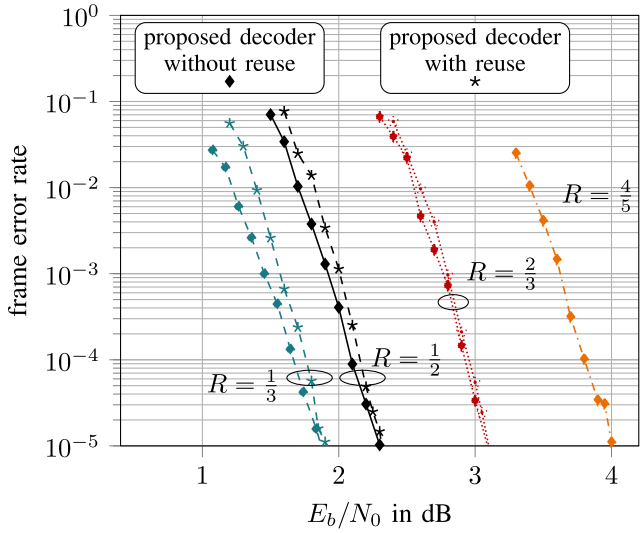


FIGURE 16. Frame error rate simulations where one static set of lookup tables is used for all rates.

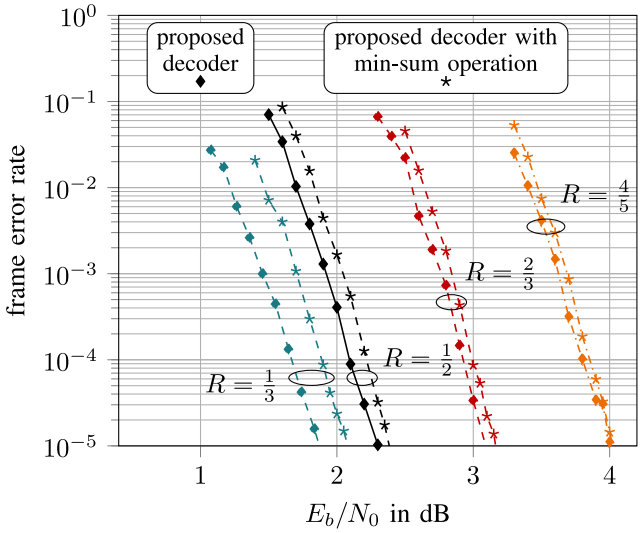


FIGURE 17. Frame error rate simulations where all lookup tables in the check nodes are replaced by the min-sum update rule (see Figure 4(e)).

can be reduced by a factor 3 for the check nodes and by a factor of approximately 1.8 for the variable nodes. Beside the reduction in memory, the table reuse allows for more efficient implementations as the same set of lookup tables can also be used for multiple code rates. Please note that the memory requirements given in Table 3 hold only for decoder implementations on a digital signal processor or software defined radio where the lookup tables are stored in memory. In general, the mappings could also be efficiently implemented as a static logic syntheses on a FPGA or ASIC [16].

E. IMPLEMENTING THE LOOKUP TABLES

As described in Section II, the general aim of the information bottleneck is to obtain a mapping $t^{\text{out}} = f(\mathbf{t}^{\text{in}})$ which preserves the relevant information. Typically, these mappings depend on the code rate and iteration. Figure 15 shows the check node lookup tables in the last step of the cascaded structure for a code rate $R_c = 0.5$ and different iterations, i.e., iteration 1, 50 and 100. It can be observed that tables change over the iterations.

In general, the lookup table implementation shown in this paper is just one way to realize the mapping. However, the general design concepts proposed in this paper are crucial for any implementation of the learned function $f(\mathbf{t}^{\text{in}})$. In literature, threshold-based implementations are proposed which require computations in a so-called computational domain at a higher internal resolution as compared to conventional information bottleneck decoder [29]. In contrast, at least for the check node, the min-sum operation can be performed using the integer-valued cluster indices as pointed out in Section II and proposed in [24]. Simulation results for such a hybrid approach where the lookup tables replace the variable node operation but the check node performs the min-sum operation (see Figure 4(e)) are shown in Figure 17. In turn, the mapping at the check node is fixed for all iterations and does not change. However, the variable node mappings are still adapted to the evolving densities and, thus, change in each iteration. Again, only the results for the proposed decoder with the parameters from Table 1 are shown as reference, for the sake of clarity. It can be observed that the performance of the proposed decoder with the min-sum update rule of Figure 4(e) is much better than the state-of-the-art offset min-sum decoder (see Figure 11 and Figure 12), especially for the lowest rate, i.e., $R_c = 1/3$. For example, at a FER of 10^{-4} the offset min-sum decoder shown in Figure 12 is outperformed by 0.6 dB. This is a very interesting observation as the 4 bit min-sum decoders are typically known to work fairly bad for low code rates. Only if the resolution is reduced further, e.g., down to 3 bit, the hybrid approach with the min-sum operation at the check node shows an early error floor.

VIII. CONCLUSION

This paper uses the information bottleneck method to efficiently represent reliability information, reducing the data transfer and computational complexity of protograph-based raptor-like LDPC decoding. The proposed decoder extends the information bottleneck decoder design to incorporate puncturing and leverages the inherent rate-compatibility of this powerful class of LDPC codes to develop a rate-compatible decoder. The proposed information bottleneck framework integrates a message alignment module into the decoder design to dynamically adjust to the degree distribution for various rates. This approach accommodates puncturing for all supported rates without significantly increasing the number of required information bottleneck lookup tables. It was shown that lookup tables can be reused for various rates due to the code structure which drastically reduces the memory demand and the implementation complexity in a rate-compatible decoder. The proposed

information bottleneck decoder exchanges only coarsely quantized messages and replaces the arithmetic in the node operations by lookup tables. This decoder performs only 0.2 dB worse than the belief-propagation algorithm and outperforms the offset-min-sum algorithm. The impact of the bit resolution was investigated and it was shown that a 3 bit information bottleneck decoder is still able to outperform an offset-min-sum decoder with larger bit resolution. Nonetheless, the main contributions of this paper are with respect to the general mutual-information-based design of the LDPC decoders and are not to limited any particular implementation of the mappings $f(\mathbf{t}^{\text{in}})$. Interestingly, the considered hybrid approach [14], containing the min-sum operation at the check node and the lookup operation at the variable node works extremely well for the entire range of code rates investigated. This approach together with the presented table reuse across various code rates allows to further reduce the implementation complexity of the proposed decoder.

ACKNOWLEDGMENT

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF, POC, or SA.

REFERENCES

[1] T. Richardson and S. Kudekar, "Design of low-density parity check codes for 5G new radio," *IEEE Commun. Mag.*, vol. 56, no. 3, pp. 28–34, Mar. 2018.
[2] C. Kestel, M. Herrmann, and N. Wehn, "When channel coding hits the implementation wall," in *Proc. IEEE 10th Int. Symp. Turbo Codes Iterative Inf. Process. (ISTC)*, ' Hong Kong, Dec. 2018, pp. 1–6.
[3] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X.-Y. Hu, "Reduced-complexity decoding of LDPC codes," *IEEE Trans. Commun.*, vol. 53, no. 8, pp. 1288–1299, Aug. 2005.
[4] D. Declercq, B. Vasic, S. K. Planjery, and E. Li, "Finite alphabet iterative decoders—Part II: Towards guaranteed error correction of LDPC codes via iterative decoder diversity," *IEEE Trans. Commun.*, vol. 61, no. 10, pp. 4046–4057, Oct. 2013.
[5] S. K. Planjery, D. Declercq, L. Danjean, and B. Vasic, "Finite alphabet iterative decoders—Part I: Decoding beyond belief propagation on the binary symmetric channel," *IEEE Trans. Commun.*, vol. 61, no. 10, pp. 4033–4045, Oct. 2013.
[6] J. K.-S. Lee and J. Thorpe, "Memory-efficient decoding of LDPC codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2005, pp. 459–463.
[7] B. M. Kurkoski, K. Yamaguchi, and K. Kobayashi, "Noise thresholds for discrete LDPC decoding mappings," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, New Orleans, LO, USA, Dec. 2008, pp. 1–5.
[8] J. Lewandowsky, M. Stark, and G. Bauch, "Optimum message mapping LDPC decoders derived from the sum-product algorithm," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2016, pp. 1–6.
[9] N. Tishby, F. C. N. Pereira, and W. Bialek, "The information bottleneck method," in *Proc. 37th Allerton Conf. Commun. Comput.*, 1999, pp. 368–377.
[10] H. Witsenhausen and A. Wyner, "A conditional entropy bound for a pair of discrete random variables," *IEEE Trans. Inf. Theory*, vol. 21, no. 5, pp. 493–501, Sep. 1975.
[11] J. Lewandowsky and G. Bauch, "Information-optimum LDPC decoders based on the information bottleneck method," *IEEE Access*, vol. 6, pp. 4054–4071, 2018.
[12] M. Stark, J. Lewandowsky, and G. Bauch, "Information-bottleneck decoding of high-rate irregular LDPC codes for optical communication using message alignment," *Appl. Sci.*, vol. 8, no. 10, p. 1884, 2018.
[13] F. J. C. Romero and B. M. Kurkoski, "LDPC decoding mappings that maximize mutual information," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 9, pp. 2391–2401, Sep. 2016.
[14] M. Meidlinger, G. Matz, and A. Burg, "Design and decoding of irregular LDPC codes based on discrete message passing," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1329–1343, Mar. 2020.
[15] X. Qu and L. Yin, "Non-uniform quantization scheme for the decoding of low-density parity-check codes with the sum-product algorithm," in *Proc. 6th Int. Conf. Electron. Inf. Emerg. Commun. (ICEIEC)*, 2016, pp. 121–125.
[16] R. Ghanaatian *et al.*, "A 588-Gb/s LDPC decoder based on finite-alphabet message passing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 2, pp. 329–340, Feb. 2018.
[17] B. Smith, F. R. Kschischang, and W. Yu, "Low-density parity-check codes for discretized min-sum decoding," in *Proc. 23rd Biennial Symp. Commun.*, 2006, pp. 14–17.
[18] S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel, "Quasi-cyclic protograph-based raptor-like LDPC codes for short block-lengths," *IEEE Trans. Inf. Theory*, vol. 65, no. 6, pp. 3758–3777, Jun. 2019.
[19] M. Stark, L. Wang, G. Bauch, and R. Wesel, "Information bottleneck decoding of rate-compatible 5G-LDPC codes," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Dublin, Ireland, Jun. 2020.
[20] N. Slonim, "The information bottleneck: Theory and applications," Ph.D. dissertation, Interdiscipl. Center Neural Comput., Hebrew Univ. Jerusalem, 2002.
[21] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," *IPN Progr. Rep.*, vol. 42, no. 154, pp. 42–154, 2003.
[22] D. Divsalar, S. Dolinar, C. R. Jones, and K. Andrews, "Capacity-approaching protograph codes," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 876–888, Aug. 2009.
[23] T.-Y. Chen, K. Vakilinia, D. Divsalar, and R. D. Wesel, "Protograph-based raptor-like LDPC codes," *IEEE Trans. Commun.*, vol. 63, no. 5, pp. 1522–1532, May 2015.
[24] M. Meidlinger, A. Balatsoukas-Stimming, A. Burg, and G. Matz, "Quantized message passing for LDPC codes," in *Proc. 49th Asilomar Conf. Signals Syst. Comput.*, Pacific Grove, CA, USA, Nov. 2015, pp. 1606–1610.
[25] W. Ryan and S. Lin, *Channel Codes: Classical and Modern*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[26] M. Stark, J. Lewandowsky, and G. Bauch, "Information-optimum LDPC decoders with message alignment for irregular codes," in *Proc. IEEE Global Commun. Conf. (Globecom)*, Abu Dhabi, UAE, Dec. 2018, pp. 1–6.
[27] J. Chen and P. M. C. Fossorier, "Density evolution for BP-based decoding algorithms of LDPC codes and their quantized versions," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, vol. 2. Taipei, Taiwan, 2002, pp. 1378–1382.
[28] J. Zhang and P. M. C. Fossorier, "Shuffled belief propagation decoding," in *Proc. 36th Asilomar Conf. Signals Syst. Comput.*, vol. 1. Pacific Grove, CA, USA, 2002, pp. 8–15.
[29] X. He, K. Cai, and Z. Mei, "Mutual information-maximizing quantized belief propagation decoding of LDPC codes," 2019. [Online]. Available: arXiv:1904.06666.

MAXIMILIAN STARK (Student Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the Hamburg University of Technology (TUHH) in 2014 and 2017, respectively, where he is currently pursing the Ph.D. degree. In 2016, he studied at Chalmers University of Technology during an Erasmus semester. Since 2017, he has been with the Institute of Communications, TUHH, as a Research Assistant. In 2019, he was with Nokia Bell Labs, France, as a Visiting Researcher in the area of deep learning in communications. His research is in the area of machine learning methods for communications and signal processing, with particular interest in channel coding, the information bottleneck method and coarsely quantized signal processing. He was a recipient of the Karl H. Dietze Award in 2017 for his master's thesis.

LINFANG WANG (Student Member, IEEE) received the M.Sc. degree in information and communication engineering from the Harbin Institute of Technology, Harbin, China, in 2018. He is currently pursuing the Ph.D. degree with the University of California at Los Angeles, Los Angeles. His current research interests include low-complexity LDPC decoder, coding theory, and quantization theory.

GERHARD BAUCH (Fellow, IEEE) received the Dipl.-Ing. and Dr.-Ing. degrees in electrical engineering from the Munich University of Technology (TUM) in 1995 and 2001, respectively, and the Diplom-Volkswirt degree (master's in economics) from FernUniversität Hagen in 2001. In 1996, he was with the German Aerospace Center (DLR), Oberpfaffenhofen, Germany. From 1996 to 2001, he was member of Scientific Staff with TUM. From 1998 to 1999, he was also a Visiting Researcher with AT&T Labs Research, Florham Park, NJ, USA. In 2002, he joined DOCOMO Euro-Labs, Munich, Germany, where he has been managing the Advanced Radio Transmission Group. In 2007, he was additionally appointed as a Research Fellow of DOCOMO Euro-Labs. He was a Full Professor with the Universität der Bundeswehr Munich from 2009 to 2012. Since October 2012, he has been the Head of the Institute of Communications, Hamburg University of Technology.

RICHARD D. WESEL (Fellow, IEEE) received the B.S. and M.S. degrees in electrical engineering from the Massachusetts Institute of Technology in 1989, and the Ph.D degree in electrical engineering from Stanford University in 1996. He is a Professor with the Electrical and Computer Engineering Department, UCLA, and an Associate Dean for Academic and Student Affairs for the Henry Samueli School of Engineering and Applied Science, UCLA. His research is in the area of communication theory with particular interest in low-density parity-check coding, short-blocklength communication with feedback, and coding for storage. He has received the National Science Foundation CAREER Award, the Okawa Foundation Award for research in information theory and telecommunications, and the Excellence in Teaching Award from the Samueli School of Engineering. He has served as an Associate Editor for Coding and Coded Modulation for the IEEE TRANSACTIONS ON COMMUNICATIONS.