Information Bottleneck Decoding of Rate-Compatible 5G-LDPC Codes

Maximilian Stark, Gerhard Bauch
Hamburg University of Technology
Institute of Communications
{maximilian.stark,bauch}@tuhh.de

Linfang Wang, Richard D. Wesel
University of California, Los Angeles
Department of Electrical and Computer Engineering
{lfwang,wesel}@ucla.edu

Abstract—The new 5G communications standard increases data rates and supports low-latency communication that places constraints on the computational complexity of channel decoders. 5G low-density parity-check (LDPC) codes have the so-called protograph-based raptor-like (PBRL) structure which offers inherent rate-compatibility and excellent performance. Practical LDPC decoder implementations use message-passing decoding with finite precision, which becomes coarse as complexity is more severely constrained. Performance degrades as the precision becomes more coarse. Recently, the information bottleneck (IB) method was used to design mutual-information-maximizing lookup tables that replace conventional finite-precision node computations. The IB approach exchanges messages represented by integers with very small bit width. This paper extends the IB principle to the flexible class of PBRL LDPC codes as standardized in 5G. The extensions include puncturing and rate-compatible IB decoder design. As an example of the new approach, a 4-bit information bottleneck decoder is evaluated for PBRL LDPC codes over a typical range of rates. Frame error rate simulations show that the proposed scheme outperforms offset min-sum decoding algorithms and operates very close to double-precision sum-product belief propagation decoding.

I. INTRODUCTION

Low-density parity-check (LDPC) codes are used in the current 5G standard based on their powerful error-correction performance [1]. The theoretically achievable performance under message passing decoding requires high-precision message representations and computationally complex node operations. Such implementations introduce impractical latency and high power consumption. The desired throughput and latency promised by 5G [1] require practical hardware implementations that use finite-precision message passing algorithms and node computations that are simplified by smart approximations. Still, the error-rate performance of such finite-precision decoders deteriorates significantly with decreasing precision [2].

To address the challenge of good performance with low precision and simple computation, the *information bottleneck* (IB) decoder [2]–[6] combines ideas from information theory and machine learning. IB decoders differ from conventional finite-precision decoders significantly. First, instead of executing any conventional arithmetic exactly or approximated, the node operations are replaced by *relevant-information-maximizing* functions which map discrete input messages onto discrete output messages.

While similar in operation to the lookup tables developed for finite-alphabet iterative decoding (FAID) approach [7], [8], the tables used in IB decoders are learned in an unsupervised manner, using the IB method [2]–[6].

As with the FAID approach, in the entire decoder no log-likelihood ratios (LLRs) are processed at any time. Instead, integer-valued messages, sometimes called *cluster indices*, are exchanged. However, whereas the FAID approach is mainly restricted to regular LDPC codes with variable node degree three, in our previous work [4], [9], [10], IB decoders with only 4 bits of precision perform within 0.1 dB of double precision belief-propagation for arbitrary regular and arbitrary irregular LDPC codes without puncturing. For irregular codes, message alignment provides a common representation across nodes with different degrees [5]. In [11] it was shown that, with a similar decoder, a throughput up to 500 Gb/s is possible with high energy and area efficiency.

To the best of our knowledge, all information bottleneck decoders in literature are tailored for a specific rate. However, in practical systems a rate-compatible decoding scheme is favorable. Recently, so-called protograph-based raptor-like (PBRL) LDPC codes were shown to pair very powerful error-correcting capabilities and an efficient structure which enables an inherent rate-compatibility [12].

This paper presents a generalized design of IB decoders to enable decoding of 5G-related PBRL codes with a bit-width down to 4 bits while incorporating puncturing and hence rate-compatibility into the IB decoder itself. In detail, the paper contains the following main contributions:

- This paper extends the design of IB LDPC decoders from [4], [5] to include puncturing in both the high-rate mother code and the degree-one variable nodes of PBRL codes.
- This paper reframes message alignment as its own IB problem, facilitating designs for irregular LDPC codes.
- Using the new approach, a 4-bit information bottleneck decoder for a PBRL code family outperforms a 6-bit offset-min-sum decoder and performs within 0.2 dB of double precision belief propagation decoding.

Organisation: The IB method and PBRL LDPC codes are briefly reviewed in Section II. In Section III, we summarize
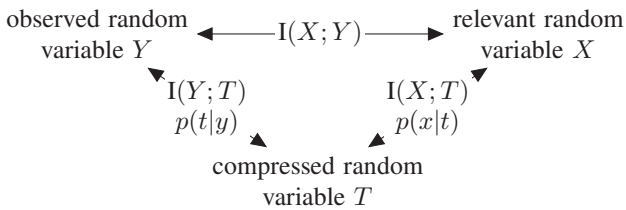
Fig. 1. Information Bottleneck setup, where $\mathrm{I}(X;T)$ is the relevant information, $\mathrm{I}(X;Y)$ is the original mutual information and $\mathrm{I}(Y;T)$ is the compression information.

the design of IB LDPC decoders. Thereafter, we use *message alignment* to incorporate puncturing. Finally, this paper targets the problem of rate-compatible decoding architectures in Section IV. In Section V, numerical simulations comparing the performance of our proposed decoder with several reference systems are provided. Section VI concludes the paper.

Notation: The realizations $y \in \mathcal{Y}$ from the sample 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 and conditional distributions are denoted $p(x,y)$ and $p(x|y)$.

II. PREREQUISITES

This section briefly reviews the information bottleneck method. Furthermore protograph-based raptor-like (PBRL) LDPC codes are introduced.

A. The Information Bottleneck Method

The information bottleneck method [13] is a mutual-information-maximizing clustering framework from machine learning. The overall information bottleneck setup is depicted in Figure 1. It considers a Markov chain $X \to Y \to T$ of three random variables. X is termed the relevant variable, Y is termed the observation and T is a compressed representation of Y. The compression is described by the conditional distribution $p(t|y)$. This compression mapping is designed such that the mutual information $\mathrm{I}(X;T)$ is maximized while at the same time the mutual information $\mathrm{I}(Y;T)$ is minimized. If the mapping $p(t|y)$ uniquely assigns a t to each y with probability 1, this mapping can be implemented in a lookup table such that $t = f(y)$. Algorithms to find suitable compression mappings are described in [14]. These algorithms require the joint distribution $p(x,y)$ and the desired cardinality $|\mathcal{T}|$ of the compression variable T as inputs.

B. Protograph-Based Raptor-Like (PBRL) LDPC Codes

Thorpe [15], [16] 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 [12], [17]. The protograph of an PBRL LDPC code consists of two parts: (1) a highest-rate code (HRC) protograph and (2) an incremental redundancy code (IRC) protograph. Thereby, the
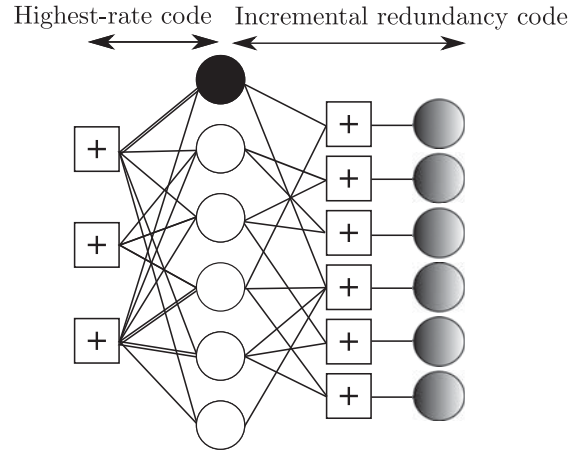

Fig. 2. Protograph of a PBRL LDPC code with a puntured variable node in the highest-rate code (shaded node) and degree-one variable nodes being possibly punctured to adapt the rate (partial shade).

IRC provides lower rates as more of its variable nodes are transmitted, starting from the top. A more detailed introduction to PBRL LDPC codes is given in [12], [17].

This paper addresses the issue of designing IB decoders that accommodate the puncturing that is inherent to PBRL code families. As pointed out in [17], one or two variable nodes in the HRC are typically punctured, 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 HRC, 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 DECODING OF LDPC CODES

In the following section, this paper introduces all the required steps to construct an information bottleneck decoder as described in [4] and [5].

A. Transmission Scheme and Channel Output Quantization

We consider binary LDPC codewords transmitted over a symmetric additive white Gaussian noise (AWGN) channel with binary phase shift keying modulation (BPSK) and quantized channel outputs. We use x to denote the equally likely transmitted symbols, which serve as channel inputs. The binary channel input x and continuous channel output y are related by the transition probability $p(y|x)$. Feeding $p(y,x)$ into the information bottleneck algorithm yields the quantizer mapping $p(t_{ch}|y)$, where $t_{ch} \in \mathcal{T}_{ch}$ denotes the discrete channel output. Such a mapping is illustrated in Figure 3. In general, a representative log-likelihood ratio can be assigned to each quantization region. These representatives correspond to the quantized channel knowledge which serves as input
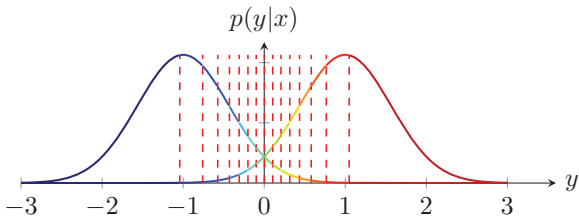
Fig. 3. Quantization boundaries for the BI-AWGN channel computed using the information bottleneck algorithm from [4].

Fig. 4. The message alignment problem posed as an Information Bottleneck, where $I(X;Z)$ is the relevant information and $I(Y;T^{out},D)$ is the compressed information.

for belief-propagation decoding. In contrast, an information bottleneck decoder does not use any quantized LLRs, but processes a single cluster index $t_{ch} \in \{1,\ldots,|\mathcal{T}_{ch}|\}$ instead.

B. Information Bottleneck Decoders for Unpunctured Binary LDPC Codes

In recent work [2]–[6], information bottleneck (IB) decoders were shown to handle the trade-off between low implementation complexity and near-optimal performance very well. When constructing an IB decoder, node operations are optimized specifically for the available discrete input and output alphabets rather than as approximations of the original belief propagation operations that assume the uncountable alphabet of all reals for inputs and outputs. The IB operations are essentially lookup tables mapping each possible set of incoming, discrete messages onto a discrete outgoing message. As a result, only highly informative integer-valued messages are passed along the edges of a Tanner graph.

To construct the discrete node operations, the joint probability distribution of the observed random variable and relevant random variable are required (cf. Figure 1). In the context of LDPC decoder design, the observed random variables are the M incoming discrete messages $\mathbf{t}^{in} = [t_1^{in},\ldots,t_M^{in}]^{\mathrm{T}}$ and the relevant random variable X depends on the node type. For a variable node, X represents the underlying code bit of a particular node, whereas, if the mapping is designed for a check node, X represents the (mod 2)-sum of the connected, possibly different code bits $b_1,\ldots,b_M$. Given the joint distribution $p(x,\mathbf{t}^{in})$ at each node type in every iteration, the information bottleneck method allows to squeeze $p(x,\mathbf{t}^{in})$ through a compact bottleneck. Meaning that the high-dimensional discrete observation vector $\mathbf{t}^{in}$, is mapped onto a scalar integer-valued cluster index $t^{out} \in \mathcal{T} = \{1,\ldots,|\mathcal{T}|\}$ defined by the mapping $p(t^{out}|\mathbf{t}^{in})$. However, the information bottleneck method aims to preserve all relevant information such that $I(X;T^{out}) \approx I(X;\mathbf{T}^{in})$. Hence, at any time, X can be inferred very precisely using $p(x|t^{out})$ (cf. Figure 1). However, once the mappings are found, the actual decoding simplifies to lookups in offline generated tables, which map the sequence of incoming integers $\mathbf{t}^{in}$ onto an outgoing integer-valued message t^{out}. Therefore, instead of passing the meaning $p(x|t^{out})$, e.g. as LLR, only cluster indices are passed which are never converted to an explicit LLR representation.

C. Message Alignment for Irregular LDPC Codes

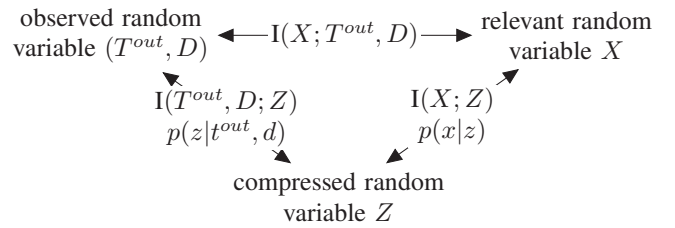In contrast to regular LDPC codes, irregular LDPC codes are characterized by nodes with varying degrees, i.e., the number of incoming messages differs. This paper leverages the edge-degree distribution [18]:

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

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. Information bottleneck decoders pass integer-valued cluster indices t instead of log-likelihood ratios. As the eventspace of $\mathcal{T}$ is *independent* of the node degree d but in contrast the particular meaning $p(x|t^{out},d)$ *depends* on the node degree, a receiving node cannot resolve if the message originates from a conveying node with high or low degree from the cluster indices alone. As the variety of node degrees increases also the range of the different meanings increases likewise. In turn, the decoder cannot exploit the full error-correction capabilities of the LDPC code because it cannot uniquely recover the actual meaning of the received message. In [5], it was revealed that with a processing at the origin nodes called *message alignment*, information bottleneck decoders can achieve competitive performance for LDPC codes with non-optimized irregular degree distributions. As an extension to the original mutual-information based lookup table design, message alignment considers the three random variables T^{out}, X and D jointly, i.e, the clusters, the channel input (which is the relevant random variable), *and* the node degree. Given the node dependent meanings $p(x,t^{out}|d)$ and the edge degree distributions, the task is to find a mapping $p(z|t^{out},d)$ such that $I(X;T^{out},D) \approx I(X;Z)$ [5]. Using this modified objective, message alignment ensures that identical outgoing cluster indices of nodes with different degrees correspond to similar beliefs on the code bits. In fact, Figure 4 reveals that the objective of message alignment is closely related to the information bottleneck setup from Figure 1. Leveraging the information bottleneck notation, one would refer to X as the relevant quantity and T^{out} *and* D represent the observation which squeeze through a compact bottleneck resulting in Z which is a compressed version of T^{out} and D but preserves the maximum relevant information $I(X;Z)$.

IV. Information Bottleneck PBRL LDPC Decoders

To decode PBRL LDPC codes the respective decoders must support puncturing. Although puncturing itself is a fairly easy problem for conventional decoders, the existing design of information bottleneck decoders prohibits the use of puncturing.
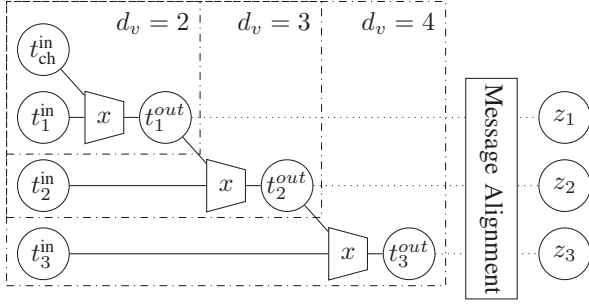
Fig. 5. Information bottleneck graph for a concatenated lookup table for $d_v = 4$ with message alignment.

This section contains our main contributions. First, this paper shows how to incorporate punctured nodes using message alignment. Second, this paper devises a generalized scheme well suited for the structure of PBRL codes enabling rate-compatibility.

A. Constructing Information Bottleneck Decoders for Punctured PBRL LDPC Codes

To increase the code rate, "punctured" codeword bits are not transmitted and are thus unknown to the receiver. In conventional approaches this corresponds to an LLR=0 for the respective punctured bits.

Going back to Figure 3 reveals that the information-optimum channel quantizer is fully symmetric [19]. Typically, the resolution of the analog-to-digital converter (ADC) is a power of two and thus even. In this case, the LLR=0 can not be represented properly. Usually the quantization boundaries are then shifted to be able to represent LLR = 0.

In information bottleneck decoders, LLR representations are ignored. Furthermore, the decoder is designed using density evolution and thus requires the complete knowledge of the statistics of the decoder input. Hence, constructing an information bottleneck decoder for puncturing is far less straightforward than in conventional decoders. In the following we propose a generic extension to include puncturing which leverages the message alignment technique.

Without loss of generality, we consider a variable node with degree $d_v = 4$ which processes one channel message and three messages received from connected check nodes to generate extrinsic information about the underlying code bit. Figure 5 shows this processing as a concatenation of two-input-lookup tables. In Figure 5 each lookup table is depicted as trapezoid with the input vector $\mathbf{t}^{in} = [t_{ch}^{in}, t_1^{in}]^{\mathrm{T}}$ or $\mathbf{t}^{in} = [t_{i-1}^{in}, t_i^{in}]^{\mathrm{T}}$ and output t_i^{out}. The respective mappings are found using the information bottleneck method. The joint distribution in the first stage, which is processing the input vector $\mathbf{t}^{in} = [t_{ch}^{in}, t_1^{in}]^{\mathrm{T}}$ is computed as follows (See [4] for details.):

$$p(x, [t_{ch}^{in}, t_1^{in}]^{\mathrm{T}}) = \frac{1}{p(x)} p(x, t_{ch}^{in}) p(x, t_1^{in}). \qquad (2)$$

Clearly, $p(x, [t_{ch}^{in}, t_1^{in}]^{\mathrm{T}})$ and thus also $p(x|t_1^{out})$, which is used in the next step, depend on the statistics of the quantized
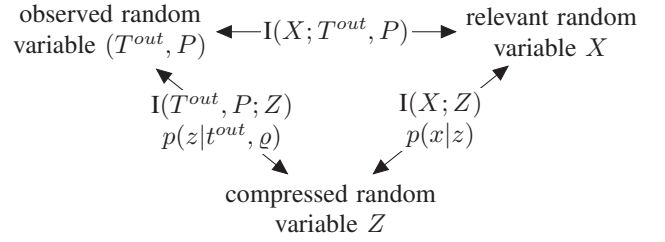

Fig. 6. Considering puncturing as message alignment problem, where $\mathrm{I}(X; Z)$ is the relevant information and $\mathrm{I}(Z; T, P)$ is the compressed information.

channel output (cf. Section III-A). When incorporating puncturing, $p(x, t_{ch}^{in})$ differs depending on whether the variable node is punctured or not. First, we introduce the random variable P with event space $\varrho \in \{$true, false$\}$ indicating if a node is punctured or not. In this paper, the puncturing rate indicates the fraction of variable nodes with degree $d > 1$ that are punctured. In the next section, it will become clear why punctured variable nodes with degree $d = 1$ are ignored when computing the puncturing rate. As a result, we rewrite (2) as

$$p(x, [t_{ch}^{in}, t_1^{in}]^{\mathrm{T}}|\varrho) = \frac{1}{p(x)} p(x, t_{ch}^{in}|\varrho) p(x, t_1^{in}). \qquad (3)$$

Due to the concatenation of lookup tables as shown in Figure 5 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. This paper proposes a novel design objective which includes puncturing in the table design. In turn, P and T are considered jointly, i.e., the objective is to find a mapping $p(z|t^{out}, \varrho)$ such that $\mathrm{I}(X; T^{out}, P) \approx \mathrm{I}(X; Z)$. As depicted in Figure 6 the resulting problem is basically a message alignment setup. As a result, one creates the mapping $p(z|t^{out}, \varrho)$ 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 significantly reduces the number of distinct lookup tables.

B. Constructing Information Bottleneck Decoders for Rate-Compatible PBRL LDPC Codes

PBRL codes can operate close to the theoretical limit for a wide range of code rates [17]. The rate can be easily adapted by puncturing or transmitting the degree-one variables (cf. Figure 2). As a punctured degree-one variable node always conveys an LLR equal to zero to the connected check nodes, these check nodes and thus entire parts of the respective Tanner graph are deactivated [17], meaning that no relevant information propagates from the punctured degree-one nodes towards the inner variable nodes with very high degree (cf. Figure 2). Hence, puncturing degree-one nodes, i.e., deactivating parts of the Tanner graph changes the *effective* degree distribution $\lambda_{eff}(z) \neq \lambda(z)$ and $\rho_{eff}(z) \neq \rho(z)$. To allow information bottleneck decoders to cope with punctured degree-one nodes and rate-adaptability the effective degree distribution is of crucial importance. To determine the effective degree distribution, the following strategy is proposed:

Variable Node: If an edge carries no information, this would correspond to LLR = 0 in a conventional decoder, the effective node degree seen by the outgoing message is reduced by one.

Check Node: If already one edge which is used to generate extrinsic information contributes no information, i.e., the corresponding LLR = 0, the outgoing message will automatically carry no extrinsic information. Thus, only the fraction of edges over which extrinsic information is passed is considered for each effective degree.

The computations are performed offline given a parity check matrix for every iteration. In Figure 5, we have depicted an unfold variable node. It can be seen that in the general scheme only one tree of lookup tables for the highest node degree in the code has to be constructed. All nodes with smaller degrees can easily reuse the tables. As mentioned above, puncturing degree-one variable nodes reduce the effective degree distribution. However, in the context of the concatenated structure presented in Figure 5, this paper argues that puncturing basically only effects the depth of the lookup tree, because if a message is punctured it would not contribute any relevant information and thus can be skipped. Hence, designing a decoder for the lowest code rate, yields the maximum depth of the lookup tree. Higher code rates will reuse this architecture but require individually optimized tables. It is possible to reuse the same tables for a range of code rates, but this analysis is left to the journal version of this paper due to space limitations.

V. RESULTS AND DISCUSSION

In this section, we present and discuss results obtained performing frame error rate simulations for an exemplary PBRL LDPC code. The code was taken from [17]. The code has $K = 1032$ information bits, and is evaluated for various code rates R_c range from $R_c = 1/3$ up to $R_c = 2/3$.

We propose to construct all involved lookup tables just once for a fixed design-E_b/N_0. 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.

We consider three reference schemes to compare the performance of our decoder. 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 flooding schedule. The received samples are *not* quantized and the internal operations are additions at the variable node and box-plus at the check node. Second, we use the layered normalized min-sum algorithm (NMSA) [20], [21] with 6 bit resolution at the check node and 6 bits at the variable node. Again the inputs to the decoder are *not* quantized. The operations here are again additions at the variable nodes but the normalized min-sum approximation is used at the check nodes. Third, we use the offset-min-sum decoder with only 4 bit resolution at the check node and 6 bits 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 a fully 4 bit integer architecture. This means, starting form the channel quantizer which outputs 4
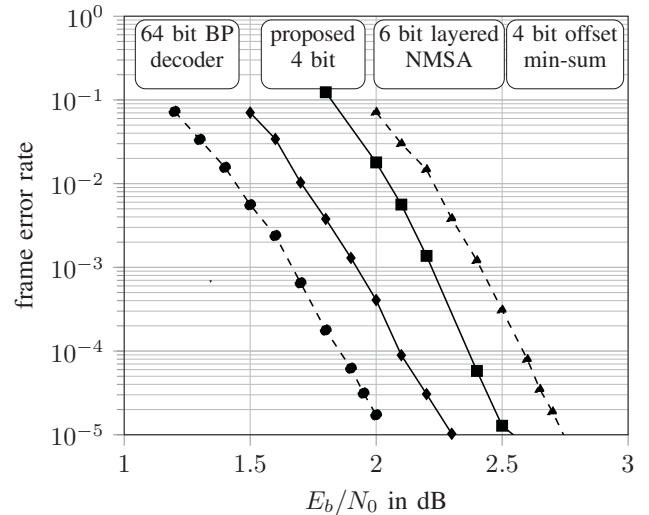

Fig. 7. Frame error rates for the proposed scheme (diamond-marker), and the reference schemes summarized in Table I for the considered PBRL LDPC code with code rate $R_c = 1/2$.
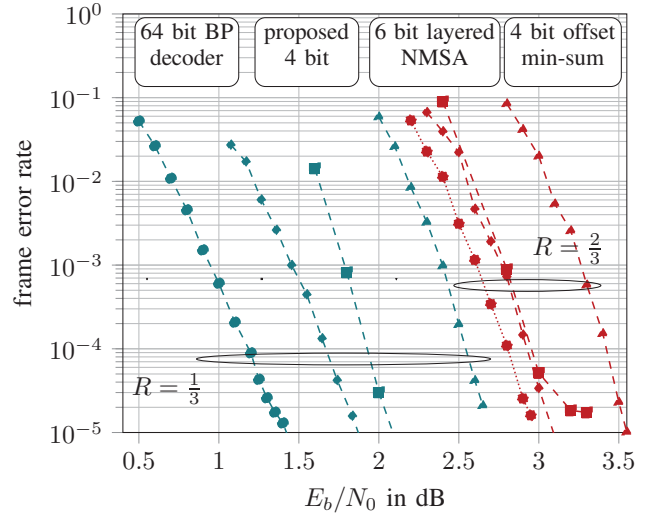

Fig. 8. Frame error rates for the proposed scheme (diamond-marker) and the reference schemes summarized in Table I for the considered PBRL LDPC code with code rate $R_c = 1/3$ (blue, dashed), $2/3$ (red, dotted).

bit integers, the internal messages require only 4 bits and only *lookup* operations performed. These lookups do not mimic any arithmetic function but realize the relevant-information preserving mappings found using the information bottleneck method.

The most important parameters of the applied decoders are summarized in Table I for a quick overview. First we consider a decoder designed for a fixed rate of $R = 0.5$. The results are shown in Figure 7. As expected the belief-propagation (BP) decoder (-marker) achieves the best frame error rate performance, but at the same time, has the highest computational complexity (cf. Table I). 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 offset-

TABLE I

SIMULATION PARAMETERS

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

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 7 we conclude that a conventional offset-min-sum decoder which exchanges only 4 bit messages can not 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 8 shows 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 outperforms the 4 bit offset min-sum decoder and the 6 bit NMSA decoder for all investigated rates.

VI. CONCLUSION

This paper uses the information bottleneck method to efficiently represent reliability information, reducing the data transfer and computational complexity of 5G protograph-based raptor-like LDPC decoding. The proposed decoder extends the information bottleneck method 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 to dynamically adjust to the degree distribution of input messages. This approach accommodates puncturing for all supported rates without significantly increasing the number of required information bottleneck lookup tables. The proposed information bottleneck decoder exchanges only 4 bit integers and replaces the arithmetic in the node operations by lookup tables. This decoder performs only 0.2 dB worse than the belief-propagation decoder and outperforms the offset-min-sum algorithm. Future work will investigate information bottleneck decoders with flexible bit widths to allow performance to be optimized based on available memory and a table reuse for several code rates.

REFERENCES

[1] T. Richardson and S. Kudekar, "Design of low-density parity check codes for 5G new radio," *IEEE Communications Magazine*, vol. 56, no. 3, pp. 28–34, March 2018.

[2] M. Meidlinger and G. Matz, "On irregular LDPC codes with quantized message passing decoding," in *Proc. SPAWC'17*. Piscataway, NJ: IEEE, 2017, pp. 1–5.

[3] J. Lewandowsky and G. Bauch, "Trellis based node operations for LDPC decoders from the information bottleneck method," in *Proc. ICSPCS'15*, Dec 2015, pp. 1–10.

[4] J. Lewandowsky and G. Bauch, "Information-optimum LDPC decoders based on the information bottleneck method," *IEEE Access*, vol. 6, pp. 4054–4071, 2018.

[5] M. Stark, J. Lewandowsky, and G. Bauch, "Information-bottleneck decoding of high-rate irregular LDPC codes for optical communication using message alignment," *Applied Sciences*, vol. 8, no. 10, 2018.

[6] 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.

[7] D. Declercq et al., "Finite alphabet iterative decodersparst ii: Towards guaranteed error correction of LDPC codes via iterative decoder diversity," *IEEE Transactions on Communications*, vol. 61, no. 10, pp. 4046–4057, October 2013.

[8] S. K. Planjery et al., "Finite alphabet iterative decodersparst i: Decoding beyond belief propagation on the binary symmetric channel," *IEEE Transactions on Communications*, vol. 61, no. 10, pp. 4033–4045, October 2013.

[9] J. Lewandowsky, M. Stark, and G. Bauch, "Information bottleneck graphs for receiver design," in *Proc. ISIT'16*, Barcelona, Spain, Jul. 2016, pp. 2888–2892.

[10] G. Bauch et al., "Information-optimum discrete signal processing for detection and decoding," in *Proc. VTC-Spring'18*, Porto, Portugal, 2018.

[11] R. Ghanaatian et al., "A 588-gb/s LDPC decoder based on finite-alphabet message passing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 2, pp. 329–340, Feb. 2018.

[12] S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel, "Quasi-cyclic protograph-based raptor-like ldpc codes for short block-lengths," *IEEE Trans. on Info. Theory*, vol. 65, no. 6, pp. 3758–3777, June 2019.

[13] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," in *Proc. 37th Allerton Conference on Communication and Computation*. IEEE, 1999.

[14] N. Slonim, "The information bottleneck: Theory and applications," Ph.D. dissertation, Hebrew University of Jerusalem, 2002.

[15] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," *IPN progress report*, vol. 42, no. 154, pp. 42–154, 2003.

[16] D. Divsalar et al., "Capacity-approaching protograph codes," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 6, pp. 876–888, 2009.

[17] T. Chen et al., "Protograph-based raptor-like LDPC codes," *IEEE Transactions on Communications*, vol. 63, no. 5, pp. 1522–1532, May 2015.

[18] W. Ryan and S. Lin, *Channel codes: classical and modern*. Cambridge University Press, 2009.

[19] B. M. Kurkoski, K. Yamaguchi, and K. Kobayashi, "Noise thresholds for discrete LDPC decoding mappings," in *Proc. GLOBECOM'18*, New Orleans, LO, USA, Dec. 2008, pp. 1–5.

[20] J. Chen, "Density evolution for BP-based decoding algorithms of LDPC codes and their quantized versions," in *Proc. IEEE GLOBECOM'02*, vol. 2. Taipei, Taiwan: IEEE, 2002, pp. 1378–1382.

[21] J. Zhang and M. Fossorier, "Shuffled belief propagation decoding," in *Proc. 36th Asilomar Conference on Signals, Systems and Computers*, vol. 1. IEEE, 2002, pp. 8–15.