Graph Codes for Dual-Parameter Barrier Channels

Yuval Ben-Hur†, Saar Stern†, Yoav Cohen and Yuval Cassuto
The Andrew and Erna Viterbi Department of Electrical and Computer Engineering
Technion - Israel Institute of Technology
Technion City, Haifa 3200003, Israel
Email: {{yuvalbh, saar.stern, yoavc}@campus, ycassuto@ee}.technion.ac.il

***Abstract*—A ternary barrier channel is a non-symmetric error model defined previously to address emerging applications. Conveniently, it admits a code-construction method comprising two binary constituent codes. In this paper we propose a decoding algorithm that decodes the two constituent codes jointly by message passing on a graph representing the two codes' parity-check constraints. The messages exchanged by the algorithm are likelihoods calculated from incoming messages, and they are derived in the paper based on the exact dependence between the binary values of the two codewords. Simulation results demonstrate that the proposed decoder has superior error-rate performance compared to prior decoding approaches.**

I. Introduction

Most modern digital information systems, such as data-storage devices, data processing units and communication systems, represent information using the binary alphabet. Binary representations naturally limit the system efficiency (e.g., information density/rate and power utilization), since they use only $Q = 2$ levels, while many next-generation devices are able to span multiple representation levels/states.

Devices working with ternary alphabets gained increasing popularity in recent years. For example, IoT biosensors generate data streams with 3 possible states [1], novel in-memory memristor-based processors employ ternary logic [2], and next-generation electrically erasable programmable read-only memories (EEPROMs) store ternary symbols [3], [4]. Ternary representations were also considered for traditional CMOS logic [5] and wide-band communication systems [6].

Many physical ternary channels do not admit complete symmetry in their input-output transitions. This inherent asymmetry implies that efficient coding over such devices requires to depart from the widely-studied symmetric error models. We study in this work a ternary error model in which transitions between the non-zero states are physically not possible (or have negligible probability). This model, called the *dual-parameter barrier channel*, is described in Fig. 1. Note that there are two error types for the barrier channel with input c and output r: either $c \neq 0$ and $r = 0$ (downward) or $c = 0$ and $r \neq 0$ (upward). We refer to these two types of errors as *downward* and *upward* errors, respectively. There aren't, however, errors such that $r \neq c$ and both $c \neq 0$ and $r \neq 0$. The studied channel is characterized by two parameters: p is the transition probability from a non-zero state to 0 (downward), and q is the transition probability from 0, split evenly between the two non-zero states (upward).

The barrier channel was addressed in [3], [4] for $q/2 = p$ and in [7] for $q = 0$. The main contribution of [3] (and its extension [4]) is a construction method for ternary codes correcting t barrier errors, using a pair of *binary* Hamming-metric constituent codes. The decoder suggested in [4] for this construction is a bounded-distance deocder, and it decodes the two binary codes sequentially to produce the decoded ternary codeword. Enhanced decoders beyond the codes' correction guarantees are presented in [8], [9], which employ deeper cooperation between the decoders of the two constituent codes. In this paper, we go beyond these prior decoders and develop a *fully-joint* iterative decoder that passes information back-and-forth between the two (binary) codes' parity checks on a bilayer Tanner graph. This allows to utilize the dependencies between the two constituent codes that exist in the code construction.Our proposed decoder builds upon a modified code construction from [9] that allows better cooperation between the codes.

After reviewing the barrier-code construction from [9], we develop the resulting code's representation as a Tanner graph [10] with *two layers of check nodes and edges*. Based on the resulting graph representation, we formulate the relations between the two constituent codes with respect to the ternary codeword they jointly induce, and use these to devise a message-passing decoding algorithm. The key contributions in this part are formal derivations of the likelihoods passed as messages between the different node types of the graph. Based on exhaustive simulations, the newly suggested decoder is shown to significantly improve block error rates (BLER) over the channel at a variety of p, q values, compared to a graph decoder that follows the prior approach of two-step decoding.

The remainder of the paper is organized as follows. Section II reviews preliminary results from prior work. Section III defines the barrier error-correcting code as a graph code and devises a corresponding message-passing decoding algorithm, deriving the various messages passed during decoding. Section IV includes simulation results of our proposed decoder. The paper concludes in section V.

†Equal contribution.

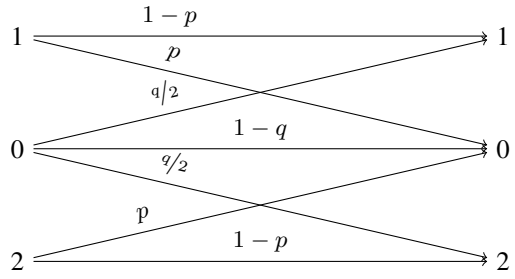

Fig. 1: The ternary dual-parameter barrier channel.

II. Preliminaries: Code construction and decoding

Error-correcting codes for various variants of the ternary barrier channel were previously constructed in [3], [4], [8], [9]. The construction used in this paper is the one from [9], which we define here for completeness. We consider ternary codes constructed based on a composition of two binary codes: an *indicator code* Θ and a *residual code* Λ. Throughout the paper, we notate $\mathbb{Z}_3 \triangleq \{0,1,2\}$. Before constructing the code, we define two useful ternary-to-binary mappings. The first is used in the construction to map a ternary codeword to its corresponding binary indicator codeword.

Definition 1: Let $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{Z}_3^n$. The indicator mapping of $\boldsymbol{x}$ is defined as $\imath(\boldsymbol{x}) \triangleq (\imath(x_1), \ldots, \imath(x_n))$ where

$$\imath(x_j) = \begin{cases} 1, & x_j \in \{1,2\} \\ 0, & x_j = 0 \end{cases} . \tag{1}$$

The second mapping is used in the construction to map a ternary codeword to its corresponding binary residual codeword.

Definition 2: Let $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{Z}_3^n$. The residual mapping of $\boldsymbol{x}$ is defined as $\psi(\boldsymbol{x}) \triangleq (\psi(x_1), \ldots, \psi(x_n))$ where

$$\psi(x_j) = \begin{cases} 1, & x_j = 2 \\ 0, & x_j \in \{0,1\} \end{cases} . \tag{2}$$

The ternary barrier code is defined as follows [9].

Definition 3 (Barrier code): Let $\Theta \subseteq \mathbb{Z}_2^n$ and $\Lambda \subseteq \mathbb{Z}_2^n$ be binary codes and let $n \in \mathbb{N}$. Then, every codeword $\boldsymbol{c} = (c_1, \ldots, c_n)$ in the ternary code $\mathcal{C} = \Theta \otimes \Lambda \subseteq \mathbb{Z}_3^n$ satisfies

1) There exists $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n) \in \Theta$ such that $\imath(\boldsymbol{c}) = \boldsymbol{\theta}$.
2) There exists $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n) \in \Lambda$ such that $\psi(\boldsymbol{c}) = \boldsymbol{\lambda}$.

The code $\mathcal{C}$ was proved [9] to have guaranteed correction capabilities over barrier channels, depending on the properties of the constituent codes Θ and Λ. In this paper, we focus on **linear** indicator and residual codes. We therefore denote their dimensions k_I and k_R, and their parity-check matrices as $\boldsymbol{H}^{(\Theta)}$ and $\boldsymbol{H}^{(\Lambda)}$, respectively.

Given a channel output $\boldsymbol{y} = (y_1, \ldots, y_n)$, a bounded-distance decoder first proposed in [4] (for a different but related construction) employs a two-step decoding scheme: first, a decoder for Θ is invoked for $\imath(\boldsymbol{y})$ to produce $\hat{\boldsymbol{\theta}}$; then, an *erasure decoder* for Λ is invoked for $\hat{\boldsymbol{\theta}} \cdot \psi(\boldsymbol{y})$, where locations in which $\hat{\theta}_j = 1$ and $y_j = 0$ are substituted with erasures; the output of this decoder is denoted $\hat{\boldsymbol{\lambda}}$. Finally, the decoded constituent codewords are combined into the decoded codeword $\hat{\boldsymbol{c}} = \hat{\boldsymbol{\theta}} + \hat{\boldsymbol{\lambda}}$.

Example 1: Let $\boldsymbol{\theta} = 001101$ and $\boldsymbol{\lambda} = 000101$, and consequently $\boldsymbol{c} = 001202$. Assume the channel induced an upward error in the 2nd symbol and a downward error in the 3rd symbol such that $\boldsymbol{y} = 020202$. Assuming the indicator code can correct 2 errors, the two-step decoder first decodes $\imath(\boldsymbol{y}) = 010101$ to $\hat{\boldsymbol{\theta}} = 001101$. The upward error is corrected (to 0), but the 3rd symbol is still ambiguous (1 or 2). Hence $\boldsymbol{y}^{res} = 00?101$. The residual decoder, therefore, receives $00?101$ as input and decodes the single erasure to $\hat{\boldsymbol{\lambda}} = 000101$. Finally, the decoded codeword is $\hat{\boldsymbol{c}} = \hat{\boldsymbol{\theta}} + \hat{\boldsymbol{\lambda}} = 001202$.

In [8] this two-step decoding scheme is refined using a list decoder for Θ, and in [9] the list of Θ-codewords is obtained adaptively by flipping bits that violate many parity-check constraints of Λ. In this paper, for the first time we propose a *full joint decoder* that passes likelihood information back-and-forth between the indicator and residual parity-check constraints. To see an illustration of the potential power of fully joint decoding, suppose the residual decoder finds that with high probability $\hat{\lambda}_j = 1$ at some position j. This means that with high probability $\hat{c}_j = 2$, and consequently also $\hat{\theta}_j = 1$. Thus passing this information to the indicator decoder can help finding the correct $\boldsymbol{\theta}$.

III. Graph decoding of Barrier codes

Any linear block code with parity-check matrix $\boldsymbol{H}$ of size $(n-k) \times n$ can be represented by a Tanner graph, i.e., a bi-partite graph $\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$ in which $\mathcal{U} = \{u_i\}_{i=1}^{n}$, $\mathcal{V} = \{v_i\}_{i=1}^{n-k}$ and there exists an edge $e_{i,j}$ between u_i and v_j if and only if $\boldsymbol{H}_{i,j} = 1$. The Tanner graph representation also allows to decode a given channel output by applying graph decoding algorithms. In this section, we represent a barrier code using a bilayer graph, as illustrated in Fig. 2, and derive a corresponding message-passing decoding algorithm.

A. Bilayer Tanner graph

When the constituent binary codes Θ and Λ are linear, it is possible to represent a barrier code as a bilayer graph: one layer of check nodes representing the parity constraints of Θ, and another layer for those of Λ. The variable nodes in this graph represent the codeword (ternary) symbols, and they are shared by both layers. Note however that this structure does *not* imply that the barrier code $\mathcal{C}$ is linear as a ternary code.

Definition 4: A bilayer graph $\mathcal{G} = (\mathcal{U}, \mathcal{E})$ is a graph for which $\mathcal{U} = \mathcal{U}^{(1)} \cup \mathcal{U}^{(2)} \cup \mathcal{U}^{(3)}$, where $\mathcal{U}^{(1)}$,$\mathcal{U}^{(2)}$ and $\mathcal{U}^{(3)}$ are disjoint sets of nodes, and $\mathcal{E} = \mathcal{E}^{(1)} \cup \mathcal{E}^{(2)}$ where

- $v \in \mathcal{U}^{(1)}$ and $u \in \mathcal{U}^{(2)}$ for any $e = (v,u) \in \mathcal{E}^{(1)}$
- $u \in \mathcal{U}^{(2)}$ and $w \in \mathcal{U}^{(3)}$ for any $e = (u,w) \in \mathcal{E}^{(2)}$

In other words, a bilayer graph contains 3 disjoint sets of nodes, $\mathcal{U}^{(1)}$,$\mathcal{U}^{(2)}$ and $\mathcal{U}^{(3)}$, where the edges connect only nodes belonging to sets with consecutive indices (i.e., $\mathcal{U}^{(1)}$ and $\mathcal{U}^{(2)}$, or $\mathcal{U}^{(2)}$ and $\mathcal{U}^{(3)}$).

Definition 5: Let $\mathcal{G} = (\mathcal{U}, \mathcal{E})$ be a graph. For a node $u \in \mathcal{U}$ and a subset of nodes $\mathcal{U}' \subseteq \mathcal{U}$, define the $\mathcal{U}'$-neighborhood of u as

$$\mathcal{N}_{\mathcal{U}'}(u) = \{v \mid (u, v) \in \mathcal{E} \text{ and } v \in \mathcal{U}'\}. \tag{3}$$

When $\mathcal{U}' = \mathcal{U}$, we denote $\mathcal{N}_{\mathcal{U}}(u)$ simply as $\mathcal{N}(u)$.

Given $\mathcal{C} = \Theta \otimes \Lambda$, where $\boldsymbol{H}^{(\Theta)}$ and $\boldsymbol{H}^{(\Lambda)}$ are parity-check matrices of Θ and Λ (respectively), we can represent $\mathcal{C}$ using a bilayer graph as follows.

Definition 6: Let $\mathcal{C}$ be a barrier code constructed as in Definition 3. Define $\mathcal{U}^{(1)} = \{v_i\}_{i=1}^{n-k_I}$, $\mathcal{U}^{(2)} = \{u_i\}_{i=1}^{n}$ and $\mathcal{U}^{(3)} = \{w_i\}_{i=1}^{n-k_R}$ as sets of nodes. The bilayer Tanner graph of $\mathcal{C}$ is defined as $\mathcal{G} = \left(\mathcal{U}^{(1)} \cup \mathcal{U}^{(2)} \cup \mathcal{U}^{(3)}, \mathcal{E}^{(1)} \cup \mathcal{E}^{(2)}\right)$ where

$$\mathcal{E}^{(1)} = \left\{(u_i, v_j) : u_i \in \mathcal{U}^{(2)}, v_j \in \mathcal{U}^{(1)}, \boldsymbol{H}^{(\Theta)}_{i,j} \neq 0\right\}$$
$$\mathcal{E}^{(2)} = \left\{(u_r, w_j) : u_r \in \mathcal{U}^{(2)}, w_j \in \mathcal{U}^{(3)}, \boldsymbol{H}^{(\Lambda)}_{i,j} \neq 0\right\}$$

We refer to the nodes $u_i \in \mathcal{U}^{(2)}$ as variable nodes, the nodes $v_i \in \mathcal{U}^{(1)}$ as indicator check nodes, and the nodes $w_i \in \mathcal{U}^{(3)}$ as residual check nodes.

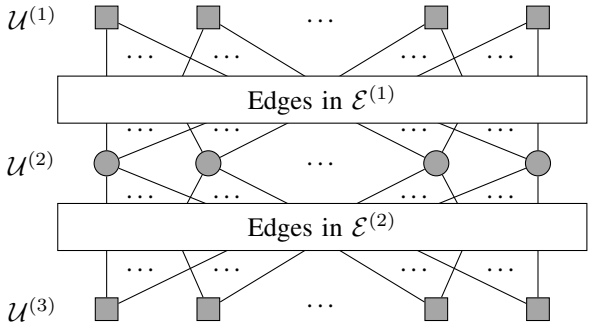

Fig. 2: An illustration of a bilayer Tanner graph for a barrier code.

In a standard Tanner graph for binary codes, each check node enforces its variable-node neighbors to have their modulo-2 sum equal 0. To use a similar property in our bilayer graph, we first apply the ternary-to-binary mappings of Definitions 1,2 on the received symbol of variable-node u_i: the *indicator mapping* $\imath(u_i)$ for edges $e \in \mathcal{E}^{(1)}$, and the *residual mapping* $\psi(u_i)$ for edges $e \in \mathcal{E}^{(2)}$. Then each indicator check node $v \in \mathcal{U}^{(1)}$ enforces the constraint

$$\sum_{u_j \in \mathcal{N}_{\mathcal{U}^{(2)}}(v)} \imath(u_j) = 0,$$

and overall the check nodes of $\mathcal{U}^{(1)}$ enforce property 1 of the construction in Definition 3, namely that $\boldsymbol{H}^{(\Theta)}\imath\left(\boldsymbol{c}^\top\right) = \boldsymbol{0}$. Similarly, each residual check node $w \in \mathcal{U}^{(3)}$ enforces the constraint

$$\sum_{u_j \in \mathcal{N}_{\mathcal{U}^{(2)}}(w)} \psi(u_j) = 0,$$

and overall the check nodes of $\mathcal{U}^{(3)}$ enforce property 2 of the construction in Definition 3, namely that $\boldsymbol{H}^{(\Lambda)}\psi\left(\boldsymbol{c}^\top\right) = \boldsymbol{0}$.

B. Message-passing joint decoding

Now that we have a (bilayer) Tanner-graph representation of the barrier code $\mathcal{C}$, we are ready to define a *message-passing decoding algorithm* for it. Message-passing decoding employs an iterative procedure that transfers information ("messages") over the edges of the Tanner graph. This information is used to compute the conditional probability of a codeword symbol u_i given an observed channel output y_i. In our proposed decoder, we define message-passing iterations for both the indicator edges ($\mathcal{E}^{(1)}$) and residual edges ($\mathcal{E}^{(2)}$). In those iterations, partial decoding results of the indicator (respectively, residual) code are transferred across the variable nodes to decode the residual (respectively, indicator) code.

We describe the message-passing procedure in general form, which applies both to messages passed over $\mathcal{E}^{(1)}$ (with indicator check nodes) or over $\mathcal{E}^{(2)}$ (with residual check nodes). We denote a message from node u to node v at iteration t as $m^{(t)}_{u\to v}(\cdot)$. The message carries information regarding a variable node's (i.e., node in $\mathcal{U}^{(2)}$) mapped binary value, namely its probability to be 0 ($m^{(t)}_{u\to v}(0)$) or 1 ($m^{(t)}_{u\to v}(1)$). We notate the set of messages from a set of arbitrary nodes $\mathcal{U}$ to a target node v at iteration t as $M^{(t)}_{\mathcal{U}\to v} = \left\{m^{(t)}_{u'\to v}(\cdot)\right\}_{u'\in\mathcal{U}}$.

We first present the indicator decoding sequence, and then explain how it is adapted to the residual code. For a bilayer graph $\mathcal{G} = (\mathcal{U}^{(1)} \cup \mathcal{U}^{(2)} \cup \mathcal{U}^{(3)}, \mathcal{E}^{(1)} \cup \mathcal{E}^{(2)})$, given a channel output $\boldsymbol{y} = (y_1, \ldots, y_n) \in \mathbb{Z}_3^n$, an indicator message-passing iteration is defined as follows.

1) *Check to variable*: For each edge $(v_j, u_i) \in \mathcal{E}^{(1)}$ where $v_j \in \mathcal{U}^{(1)}$, calculate

$$m^{(t)}_{v_j\to u_i}(b) = \Pr\left\{\sum_{u_k\in\mathcal{N}(v_j)} \imath(u_k) = 0 \middle| \begin{matrix}\imath(u_i) = b,\\ M^{(t-1)}_{\mathcal{U}^{(2)}\setminus u_i\to v_j}\end{matrix}\right\}. \tag{4}$$

2) *Variable to check*: For each edge $(u_i, v_j) \in \mathcal{E}^{(1)}$ where $v_j \in \mathcal{U}^{(1)}$, calculate

$$m^{(t)}_{u_i\to v_j}(b) = \Pr\left\{\imath(u_i) = b \middle| \begin{matrix}y_i,\, M^{(t)}_{\mathcal{U}^{(1)}\setminus v_j\to u_i},\\ M^{(t')}_{\mathcal{U}^{(3)}\to u_i}\end{matrix}\right\}, \tag{5}$$

where t' refers to the last iteration in which $M^{(t)}_{\mathcal{U}^{(3)}\to u_i}$ was updated. Before the first iteration, we initialize each message $m^{(0)}_{u\to v_j}(b)$, where $u \in \mathcal{U}^{(2)}$ and $v_j \in \mathcal{U}^{(1)}$, to the $\Pr\{\imath(u) = b \mid y\}$. The messages $m^{(t)}_{w_j\to u}(b)$ in $M^{(t'=0)}_{\mathcal{U}^{(3)}\to u_i}$ are initialized to $1/2$.

The sequence above can be adapted to the residual decoding iteration by changing $\mathcal{E}^{(1)}$ to $\mathcal{E}^{(2)}$, switching $\mathcal{U}^{(1)}$ and $\mathcal{U}^{(3)}$ and applying $\psi(\cdot)$ instead of $\imath(\cdot)$. We refer to these adapted terms as the *residual-modified* equations. We notate the messages in case of a residual iteration by μ instead of m (e.g., $m^{(t)}_{u\to v}$ is changed to $\mu^{(t)}_{u\to v}$).

Before we derive the messages themselves (illustrated in Fig. 3), we present our iterative message-passing algorithm that uses these messages to decode $\mathcal{C}$. The proposed algorithm comprises a maximum of T iterations, where every iteration is

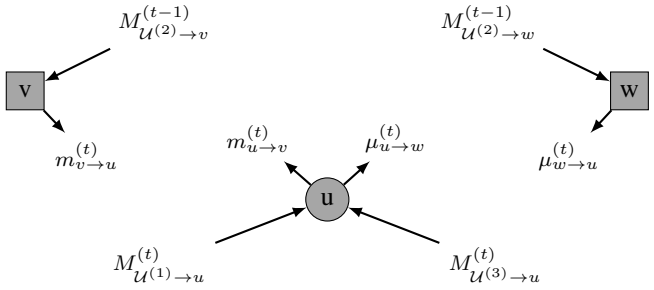

Fig. 3: Illustration of the messages accumulated and emanated from nodes in the bilayer Tanner graph, where $v \in \mathcal{U}^{(1)}, u \in \mathcal{U}^{(2)}$ and $w \in \mathcal{U}^{(3)}$.

either an indicator iteration or a residual iteration according to a predefined sequence $\boldsymbol{s} \in \{\text{'Ind','Res'}\}^T$. Before each such iteration, the decoder calculates the current ternary values that have maximal probability given the variable-nodes' incoming messages, and outputs them in case they correspond to a legal coderword.

Algorithm 1 Joint message-passing decoding of barrier codes

1: **Input:** Channel output $\boldsymbol{y} \in \mathbb{Z}_3^n$, bilayer Tanner graph $\mathcal{G}$, sequence $\boldsymbol{s}$
2: **Output:** Decoded codeword $\hat{c}$ or 'failure'
3: Initialize messages on all edges according to Section III-C
4: **for** $t = 1, \ldots, T$ **do**
5: **if** s_t equals 'Ind' **then** ▷ indicator iteration
6: Perform indicator iteration: Apply Eq. (4), then apply Eq. (5)
7: **else** ▷ residual iteration
8: Perform residual iteration: Apply residual-modified Eq. (4), then apply residual-modified Eq. (5)
9: **end if**
10: Update $\Pr\left\{u_i | y_i, M^{(t)}_{\mathcal{U}^{(1)}\to u_i}, M^{(t)}_{\mathcal{U}^{(3)}\to u_i}\right\}$ for $u_i = 0,1,2$ and $i = 1, \ldots, n$. Then, calculate $\hat{\boldsymbol{c}} = (\hat{c}_1, \ldots, \hat{c}_n)$ where

$$\hat{c}_i = \arg\max_{u_i=0,1,2} \Pr\left\{u_i | y_i, M^{(t)}_{\mathcal{U}^{(1)}\to u_i}, M^{(t)}_{\mathcal{U}^{(3)}\to u_i}\right\} \quad (6)$$

11: **if** $\hat{\boldsymbol{c}}$ is a codeword in $\mathcal{C}$ **then return** $\hat{\boldsymbol{c}}$
12: **end if**
13: **end for**
14: **return** 'failure'

C. Derivation of messages for the Barrier channel

We now derive closed-form expressions for the messages sent over the bilayer Tanner graph's edges during decoding. For ease of the derivation, we use the following indicator/residual bit probabilities, based on the previously calculated messages for each node $u_i \in \mathcal{U}^{(2)}$,

$$m^{(t)}_{u_i}(b) = \Pr\{\imath(u_i) = b \mid y_i, M^{(t)}_{\mathcal{U}^{(1)}\to u_i}, M^{(t)}_{\mathcal{U}^{(3)}\to u_i}\} \quad (7)$$

for the indicator bits, and

$$\mu^{(t)}_{u_i}(b) = \Pr\{\psi(u_i) = b \mid y_i, M^{(t)}_{\mathcal{U}^{(1)}\to u_i}, M^{(t)}_{\mathcal{U}^{(3)}\to u_i}\} \quad (8)$$

for the residual bits. Note that $m^{(0)}_{u_i}(b) = \Pr\{\imath(u_i) = b \mid y_i\}$ and $\mu^{(0)}_{u_i}(b) = \Pr\{\psi(u_i) = b \mid y_i\}$.

Instead of passing probability values, we use log-likelihood ratios (LLR) for numerical stability and mathematical simplicity.

Definition 7: Let $\mathcal{G}$ be the bilayer tanner graph of the code $\mathcal{C}$, and $m^{(t)}(\cdot)$ a message on the graph. Then the LLR is defined as $\mathcal{L}(m^{(t)}) = \ln\left(\frac{m^{(t)}(0)}{m^{(t)}(1)}\right)$, similarly for $\mu^{(t)}(\cdot)$.

As a first step, the messages $\mathcal{L}\left(m^{(0)}_{u_i\to v_j}\right)$ and $\mathcal{L}\left(\mu^{(0)}_{u_i\to w_j}\right)$, where $u_i \in \mathcal{U}^{(2)}$, $v_j \in \mathcal{U}^{(1)}$ and $w_j \in \mathcal{U}^{(3)}$, are all initialized to $\mathcal{L}\left(m^{(0)}_{u_i}\right)$ and $\mathcal{L}\left(\mu^{(0)}_{u_i}\right)$ (respectively). We now derive their explicit expressions, used in line 3 of Algorithm 1 (as probabilities instead of LLRs).

Proposition 1: Let $u \in \mathbb{Z}^3$ be a symbol transmitted through a barrier channel with parameters (p, q) and let y be the channel output. Then,

$$\mathcal{L}\left(m^{(0)}_u\right) = \begin{cases} \ln\left(\frac{1-q}{p}\right), & \text{if } y = 0 \\ \ln\left(\frac{q}{1-p}\right), & \text{if } y = 1 \\ \ln\left(\frac{q}{1-p}\right), & \text{if } y = 2 \end{cases}, \quad \mathcal{L}\left(\mu^{(0)}_u\right) = \begin{cases} \ln\left(1 + 2\frac{1-q}{p}\right), & \text{if } y = 0 \\ \infty, & \text{if } y = 1 \\ \ln\left(\frac{q}{1-p}\right), & \text{if } y = 2 \end{cases}. \quad (9)$$

Proof: Denote the prior probability $\rho_{u'} \triangleq \Pr\{u = u'\}$ for $u' = 0, 1, 2$. Note that when Θ is a linear code, $\rho_0 = \rho_1 + \rho_2 = 1/2$, and when Λ is a linear code $\rho_1 = \rho_2 = 1/4$. Hence,

$$\begin{aligned} \mathcal{L}\left(m^{(0)}_u\right) &= \ln\left(\frac{\Pr\{\imath(u) = 0 \mid y\}}{\Pr\{\imath(u) = 1 \mid y\}}\right) \\ &= \ln\left(\frac{\Pr\{y \mid u = 0\}\rho_0}{\Pr\{y \mid u = 1\}\rho_1 + \Pr\{y \mid u = 2\}\rho_2}\right) \\ &= \begin{cases} \ln\left(\frac{1-q}{p}\right), & \text{if } y = 0 \\ \ln\left(\frac{q}{1-p}\right), & \text{if } y = 1 \\ \ln\left(\frac{q}{1-p}\right), & \text{if } y = 2 \end{cases} \end{aligned} \quad (10)$$

where the last transition is according to the barrier channel

definition. Similarly, the initial residual message is

$$\mathcal{L}\left(\mu_u^{(0)}\right) = \ln\left(\frac{\Pr\{\psi(u)=0 \mid y\}}{\Pr\{\psi(u)=1 \mid y\}}\right) = \ln\left(\frac{\Pr\{y \mid u=0\}\rho_0 + \Pr\{y \mid u=1\}\rho_1}{\Pr\{y \mid u=2\}\rho_2}\right) = \begin{cases} \ln\left(1+2\frac{1-q}{p}\right), & \text{if } y=0 \\ \infty, & \text{if } y=1 \\ \ln\left(\frac{q}{1-p}\right), & \text{if } y=2 \end{cases}. \quad (11)$$

■

At each iteration t, the messages are sent from check nodes to variable nodes, and then vice-versa. Since each layer's subgraph of the bilayer Tanner graph $\mathcal{G}$ functions as a "standard" bipartite Tanner graph on its own (either for the indicator or residual code), the messages sent from check nodes to variable nodes remain as in the "standard" setting [11]. Hence, for an indicator check node $v \in \mathcal{U}^{(1)}$ and a variable node $u \in \mathcal{U}^{(2)}$, denote the set $\mathcal{N}_{\mathcal{U}^{(2)}\setminus\{u_i\}}(v_j)$, i.e., neighbors of v from $\mathcal{U}^{(2)}$ excluding u, as $\mathcal{N}_{u,v}$. Then, following a known procedure to combine incoming messages at a check node [12], we get

$$\mathcal{L}\left(m_{v\to u}^{(t)}\right) = \left(\prod_{u'\in\mathcal{N}_{u,v}} \alpha_{u',v}^{(t-1)}\right) \phi\left(\sum_{u'\in\mathcal{N}_{u,v}} \phi\left(\beta_{u',v}^{(t-1)}\right)\right), \quad (12)$$

where $\phi(x) \triangleq \ln\left(\frac{e^x+1}{e^x-1}\right)$, $\alpha_{u',v}^{(\tau)} = \text{sign}\left(\mathcal{L}\left(m_{u'\to v}^{(\tau)}\right)\right)$ and $\beta_{u',v}^{(\tau)} = \left|\mathcal{L}\left(m_{u'\to v}^{(\tau)}\right)\right|$. Similarly, the residual check to variable messages can be calculated as follows. For a residual check node $w \in \mathcal{U}^{(3)}$ and a variable node $u \in \mathcal{U}^{(2)}$, we calculate $\mathcal{L}\left(\mu_{w\to u}^{(t)}\right)$ by applying the following modifications to Eq. (12): 1) $\alpha_{u',w}^{(\tau)} = \text{sign}\left(\mathcal{L}\left(\mu_{u'\to w}^{(\tau)}\right)\right)$, 2) $\beta_{u',w}^{(\tau)} \triangleq \left|\mathcal{L}\left(\mu_{u'\to w}^{(\tau)}\right)\right|$.

The variable-to-check messages deviate from the "standard" Tanner-graph message passing, as they consider information both from the indicator check nodes and the residual check nodes, which should be combined meaningfully. We present the indicator variable-to-check messages in Proposition 2 and then the residual variable-to-check messages in Proposition 3.

Proposition 2 (variable to indicator check): Let $u \in \mathcal{U}^{(2)}$ be a variable node and let $v \in \mathcal{U}^{(1)}$ be an indicator parity-check node. Then,

$$\mathcal{L}(m_{u\to v}^{(t)}) = \mathcal{L}(m_u^{(0)}) + \sum_{v'\in\mathcal{N}_{\mathcal{U}^{(1)}\setminus v}(u)} \mathcal{L}(m_{v'\to u}^{(t)}) + \sum_{w'\in\mathcal{N}_{\mathcal{U}^{(3)}}(u)} \mathcal{T}^{(1)}\left(\mathcal{L}\left(\mu_{w'\to u}^{(t)}\right)\right), \quad (13)$$

where $\mathcal{T}^{(1)}(x) \triangleq -\ln\left(\frac{1}{2}+\frac{1}{2}e^{-x}\right)$.

Proof: See Appendix A. ■

The messages from variable nodes to residual parity-check nodes are derived in a similar fashion in the following proposition.

Proposition 3 (variable to residual check): Let $u \in \mathcal{U}^{(2)}$ be a variable node and let $w \in \mathcal{U}^{(3)}$ be a residual parity-check node. Then,

$$\mathcal{L}(\mu_{u\to w}^{(t)}) = \mathcal{L}\left(\mu_u^{(0)}\right) + \sum_{w'\in\mathcal{N}_{\mathcal{U}^{(3)}\setminus w}(u)} \mathcal{L}\left(\mu_{w'\to u}^{(t)}\right) + \sum_{v'\in\mathcal{N}_{\mathcal{U}^{(1)}}(u)} \mathcal{T}^{(3)}\left(\mathcal{L}\left(m_{v'\to u}^{(t)}\right)\right), \quad (14)$$

where $\mathcal{T}^{(3)}(x) \triangleq \ln\left(\frac{1}{3}+\frac{2}{3}e^x\right)$.

Proof: See Appendix B. ■

The function $\mathcal{T}^{(1)}$ essentially transfers information from the residual code to the indicator code. Given a variable node u, $\mathcal{T}^{(1)}$ translates a likelihood ratio

$$\frac{\Pr\{\text{satisfied residual check}|\psi(u)=0\}}{\Pr\{\text{satisfied residual check}|\psi(u)=1\}}$$

to the likelihood ratio

$$\frac{\Pr\{\text{satisfied residual check}|\imath(u)=0\}}{\Pr\{\text{satisfied residual check}|\imath(u)=1\}}.$$

For example, in the case $\mathcal{T}^{(1)}(-\infty)$ we have $\Pr\{\text{satisfied residual check}|\psi(u)=0\} = 0$, and since $\imath(u)=0$ implies $\psi(u)=0$, we have $\mathcal{T}^{(1)}(-\infty) = -\infty$ (see the left side of the solid curve in Fig. 4). Conversely, in the case $\mathcal{T}^{(1)}(\infty)$ we have $\Pr\{\text{satisfied residual check}|\psi(u)=1\} = 0$, but since $\imath(u)=1$ does *not* imply $\psi(u)=1$, $\mathcal{T}^{(1)}(\infty)$ admits a finite positive value calculated in Appendix A (see the right side of the solid curve in Fig. 4). A similar role applies to $\mathcal{T}^{(3)}$, which translates information in the reverse direction, i.e., indicator LLRs to residual LLRs (see the dashed curve in Fig. 4).

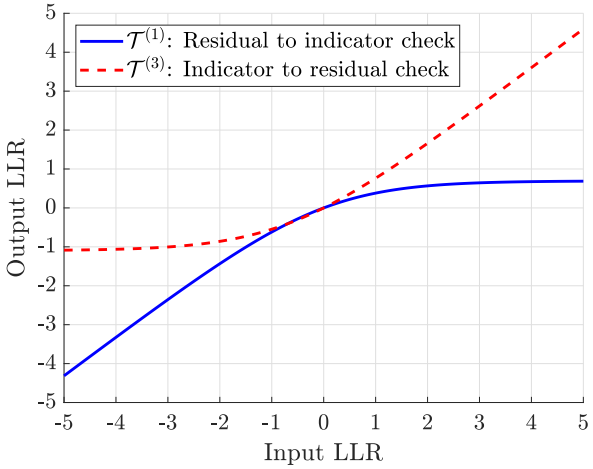

Fig. 4: Illustration of LLR functions assembling the variable to check messages of the joint decoder from Propositions 2 and 3.

To estimate the decoded symbols given the messages and the channel output, we transform the LLR messages, $\mathcal{L}(m)$ and $\mathcal{L}(\mu)$, to symbol-wise probabilities

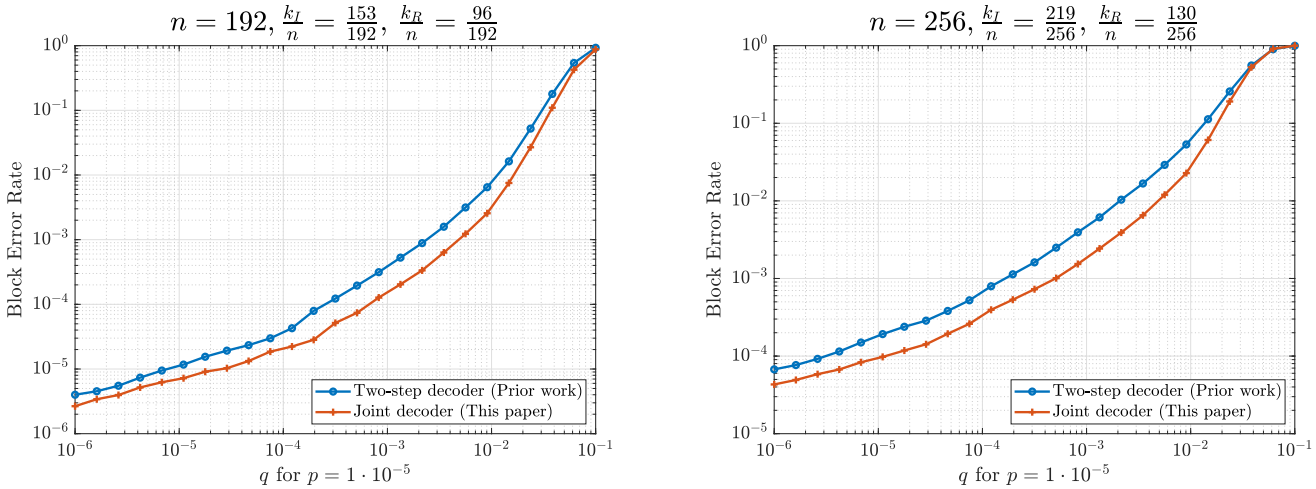

Fig. 5: Block error rates (BLER) of the suggested joint decoder and a baseline prior-work decoder simulated for $p = 10^{-5}$ with code lengths $n = 192$ (left), $n = 256$ (right).

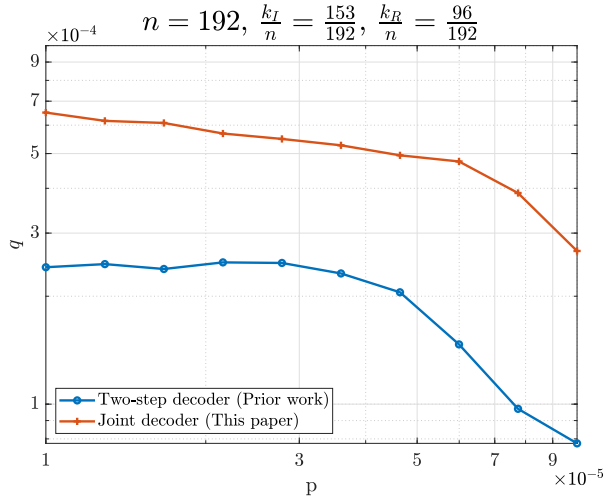

Fig. 6: The maximal achievable q obtained by the baseline and proposed decoders, as a function of p for a fixed BLER of 10^{-4}.

$\Pr\left\{u \mid y, M^{(t)}_{\mathcal{U}^{(1)}\to u}, M^{(t)}_{\mathcal{U}^{(3)}\to u}\right\}$. Writing the expression for the probability of $u = 0$ explicitly, using (7) and the fact that $\Pr\{u = 0|\cdot\} = \Pr\{\imath(u) = 0|\cdot\}$, leads to

$$\Pr\left\{u = 0|y, M^{(t)}_{\mathcal{U}^{(1)}\to u}, M^{(t)}_{\mathcal{U}^{(3)}\to u}\right\} = \frac{1}{1+\exp\left(-\mathcal{L}\left(m^{(t)}_u\right)\right)}. \tag{15}$$

Similarly, using (8) and the fact that $\Pr\{u = 2|\cdot\} = \Pr\{\psi(u) = 1|\cdot\}$, leads to

$$\Pr\left\{u = 2|y, M^{(t)}_{\mathcal{U}^{(1)}\to u}, M^{(t)}_{\mathcal{U}^{(3)}\to u}\right\} = \frac{1}{1+\exp\left(\mathcal{L}\left(\mu^{(t)}_u\right)\right)}. \tag{16}$$

Finally, $\Pr\left\{u = 1|y, M^{(t)}_{\mathcal{U}^{(1)}\to u}, M^{(t)}_{\mathcal{U}^{(3)}\to u}\right\}$ is the complement to 1 of the two previous expressions.

IV. Performance Evaluation

The evaluation of our proposed joint decoding message-passing algorithm is done using *low-density parity-check (LDPC)* codes. To construct the ternary barrier code, we use irregular binary LDPC codes with prescribed rates for both the indicator and the residual codes. After the codewords are generated at random by a designated encoder (guaranteeing the conditions of Definition 3), we pass them through the dual-parameter barrier channel. The erroneous word at the channel's output is then passed as input to the simulated decoders.

We simulated our proposed decoder from Algorithm 1, with the log likelihood ratio (LLR) messages derived in Section III-C. To facilitate a baseline decoding performance for comparison, we also simulated a sequential two-step decoder (to be detailed below), where first the indicator code is decoded on the indicator sub-graph, and then the residual code is erasure-decoded on the residual sub-graph. This decoder is similar to the one proposed in [4], only that it uses iterative decoding on the respective code sub-graphs instead of bounded-distance decoding. The baseline decoder employs the following sequence: 1) Perform 30 indicator decoding iterations on $\imath(\boldsymbol{y})$ to obtain a decoded indicator codeword $\hat{\boldsymbol{\theta}}$; 2) Generate an indicator-induced residual word $\boldsymbol{y}' = \hat{\boldsymbol{\theta}} \cdot \psi(\boldsymbol{y})$; and 3) Employ an erasure decoder on $\boldsymbol{y}'$ where locations in which $y_i = 0$ and $\hat{\theta}_i \neq 0$ are marked as erasures.

In our simulations, we focus on non-symmetric barrier channels, in which the downward error probability p is smaller than the upward error probability q. We use two code lengths: $n = 192, 256$; for the respective indicator codes we use LDPC matrices with rates $k_I/n = {}^{153}/_{192}, {}^{219}/_{256}$. For the respective residual codes we use LDPC matrices with rates $k_R/n = {}^{96}/_{192}, {}^{130}/_{256}$. The redundancy parameters chosen for the residual codes allow them to effectively assist in the decoding of the indicator codes, at a variety of channel p, q parameter combinations. We show this flexibility in our experiment presented later in the section. To avoid combinatorial

effects due to the initialization of $m_{u_i}^{(0)}(\cdot)$ and $\mu_{u_i}^{(0)}(\cdot)$ to fixed values in (9),(11), each initialization message was added with a random perturbation distributed normally with zero mean and standard deviation $0.05 \cdot m_{u_i}^{(0)}(\cdot)$ and $0.05 \cdot \mu_{u_i}^{(0)}(\cdot)$, respectively. The decoding sequence $\boldsymbol{s}$ of the joint decoder is arranged in sub-sequences of t_I indicator iterations followed by t_R residual iterations, then repeat, i.e.,

$$\boldsymbol{s} = \left[\underbrace{\text{'Ind'}, \ldots, \text{'Ind'}}_{t_I \text{ times}}, \underbrace{\text{'Res'}, \ldots, \text{'Res'}}_{t_R \text{ times}}, \ldots \right].$$

Specifically, we tested several parameter combinations empirically, and set $t_I = 6$ and $t_R = 2$ in all of our simulations. The total number of decoding iterations, i.e., the (maximal) length of $\boldsymbol{s}$, is set to 30 both for our decoder and for the baseline two-step decoder.

For both decoders we measured the *block error rate (BLER)* over large numbers of codeword and channel realizations, i.e., the fraction of transmitted codewords whose corresponding decoded codewords contained one or more symbol errors. Fig. 5 presents the results of such experiment for two codeword lengths of $n = 192$ and $n = 256$, with a fixed downward error probability $p = 10^{-5}$ and q ranging from 10^{-6} to high values approaching 1.

For all simulated scenarios, our proposed joint decoder outperforms the baseline decoder over the entire range of channel transition probabilities. Naturally, the performance of the two decoders coincides when $p + q$ is very low (few errors), or very high (too many errors), but the joint decoder presents a significant advantage in the intermediate range. The operational advantage of the proposed decoder is seen in the *horizontal* spacing between the curves: for a given output BLER, the proposed decoder can sustain a significantly higher q parameter compared to the baseline. Moreover, this gap is greater for $n = 256$ compared to $n = 192$, indicating that larger block lengths may further enjoy the advantage of joint decoding.

In Fig. 6 we fix the target BLER to 10^{-4}, and plot the maximal achievable q parameter as a function of the value of the p parameter, to meet the BLER target. The joint decoder is seen to consistently withstand larger channel error probabilities, illustrating its superiority over the prior approach of two-step decoding.

V. Conclusion

This paper considers graph codes for error correction over the dual-parameter barrier channel. Based on a novel representation using bilayer Tanner graphs, we devise a decoding algorithm that is specifically designed for the unique structure of barrier codes. Throughout the decoding process, the proposed message-passing algorithm utilizes information concerning both constituent codes mutually towards joint decoding of the ternary codeword.

A natural generalization of this work would consider barrier codes with a larger alphabet size ($Q > 3$). An additional interesting follow-up is the employment of adaptive optimization of the sequence $\boldsymbol{s}$ during decoding, rather than the fixed sequence used here.

VI. Acknowledgement

This work was supported in part by the Israel Science Foundation under grant 2525/19, and in part by the US-Israel Binational Science Foundation under grant 2023627.

References

[1] X. Tang and W. Tang, "A 151nw second-order ternary delta modulator for ecg slope variation measurement with baseline wandering resilience," in *2020 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, 2020, pp. 1–4.

[2] L. Luo, Z. Dong, X. Hu, L. Wang, and S. Duan, "Mtl: Memristor ternary logic design," *International Journal of Bifurcation and Chaos*, vol. 30, no. 15, pp. 205–222, 2020.

[3] N. Bitouzé and A. G. i Amat, "Coding for a non-symmetric ternary channel," in *2009 Information Theory and Applications Workshop*. IEEE, 2009, pp. 113–118.

[4] N. Bitouzé, A. G. i Amat, and E. Rosnes, "Error correcting coding for a nonsymmetric ternary channel," *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5715–5729, 2010.

[5] D. V. Efanov, "Ternary parity codes: Features," in *2019 IEEE East-West Design & Test Symposium (EWDTS)*. IEEE, 2019, pp. 1–5.

[6] S. Vladimirov and O. Kognovitsky, "Wideband data signals with direct ternary maximum length sequence spread spectrum and their characteristics," *Proceedings of Telecommunication Universities*, vol. 3, pp. 28–36, 2017.

[7] R. Ahlswede and H. Aydinian, "Error control codes for parallel asymmetric channels," *IEEE Transactions on Information Theory*, vol. 54, no. 2, pp. 831–836, 2008.

[8] Y. Ben-Hur and Y. Cassuto, "Coding on dual-parameter barrier channels beyond worst-case correction," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 01–06.

[9] ——, "Construction and decoding of codes over the dual-parameter barrier error model," in *2023 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2023, pp. 1136–1141.

[10] R. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on information theory*, vol. 27, no. 5, pp. 533–547, 1981.

[11] R. Gallager, "Low-density parity-check codes," *IRE Transactions on information theory*, vol. 8, no. 1, pp. 21–28, 1962.

[12] W. E. Ryan *et al.*, "An introduction to ldpc codes," *CRC Handbook for Coding and Signal Processing for Recording Systems*, vol. 5, no. 2, pp. 1–23, 2004.

Appendix A
Proof of Proposition 2

The message $m_{u \to v}^{(t)}(b)$ comprises the product of 3 terms:

1) The posterior probability of u given the channel output, denoted
$$\Pr\left\{\imath\left(u\right) = b \mid y\right\}. \tag{17}$$
2) The probability that all indicator parity checks neighboring u are satisfied
$$\prod_{v' \in \mathcal{N}_{\mathcal{U}^{(1)} \setminus v}(u)} \Pr\left\{ \sum_{u' \in \mathcal{N}(v')} \imath\left(u'\right) = 0 \middle| \begin{matrix} \imath\left(u\right) = b, \\ M_{\mathcal{U}^{(2)} \setminus \{u\} \to v'}^{(t)} \end{matrix} \right\}. \tag{18}$$
3) The probability that all residual parity checks neighboring u are satisfied
$$\prod_{v' \in \mathcal{N}_{\mathcal{U}^{(3)}}(u)} \Pr\left\{ \sum_{u' \in \mathcal{N}(v')} \psi\left(u'\right) = 0 \middle| \begin{matrix} \imath\left(u\right) = b, \\ M_{\mathcal{U}^{(2)} \setminus \{u\} \to v'}^{(t)} \end{matrix} \right\}. \tag{19}$$

Note that the first term is exactly $m_u^{(0)}(b)$ according to the initialization in Section III-C, and the second term is simply

the product of indicator check-to-variable messages from all neighbors of u excluding v, i.e., each term in the product in (18) equals $m^{(t)}_{v'\to u}(b)$. We now derive the third term.

$$\begin{aligned}&\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\psi\left(u'\right)=0\middle|\begin{matrix}\imath\left(u\right)=0,\\M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\}\\&=\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\psi\left(u'\right)=0\middle|\begin{matrix}\psi\left(u\right)=0,\\M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\}\\&=\mu^{(t)}_{v'\to u}(0).\end{aligned}\tag{20}$$

The first equality is because $\imath\left(u\right)=0$ implies $\psi\left(u\right)=0$, and the second equality is by the definition of check-to-variable messages in the residual sub-graph. Similarly, for $b=1$ we have

$$\begin{aligned}&\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\psi\left(u'\right)=0\middle|\begin{matrix}\imath\left(u\right)=1,\\M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\}\\&=\frac{\rho_1}{\rho_1+\rho_2}\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\psi\left(u'\right)=0\middle|\begin{matrix}u=1,\\M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\}\\&+\frac{\rho_2}{\rho_1+\rho_2}\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\psi\left(u'\right)=0\middle|\begin{matrix}u=2,\\M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\}\\&=\frac{\rho_1}{\rho_1+\rho_2}\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\psi\left(u'\right)=0\middle|\begin{matrix}\psi\left(u\right)=0,\\M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\}\\&+\frac{\rho_2}{\rho_1+\rho_2}\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\psi\left(u'\right)=0\middle|\begin{matrix}\psi\left(u\right)=1,\\M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\}\\&=\frac{\rho_1\mu^{(t)}_{v'\to u}(0)+\rho_2\mu^{(t)}_{v'\to u}(1)}{\rho_1+\rho_2},\end{aligned}\tag{21}$$

where the first equality is by marginalization over the two values of u corresponding to $\imath\left(u\right)=1$, the second equality is from applying the mapping $\psi\left(\cdot\right)$ to the values of u, and the last equality is by the definition of check-to-variable messages in the residual sub-graph. To calculate the corresponding LLR, recall that $\mathcal{L}$ is the natural logarithm of the division of (20) and (21) (Definition 7). Substituting $\rho_1=\rho_2={}^1\!/_4$, we get

$$-\ln\left(\frac{1}{2}+\frac{1}{2}\exp\left(-\mathcal{L}\left(\mu^{(t)}_{v'\to u}\right)\right)\right),\tag{22}$$

which equals $\mathcal{T}^{(1)}\left(\mathcal{L}\left(\mu^{(t)}_{v'\to u}\right)\right)$. We get (13) by applying Definition 7 on (17) and (18) as well.

Appendix B
Proof of Proposition 3

The message $\mu^{(t)}_{u\to v}(b)$ comprises the product of 3 terms:

1) The posterior probability of u given the channel output, denoted
$$\Pr\left\{\psi\left(u\right)=b\mid y\right\}.\tag{23}$$
2) The probability that all residual parity-checks neighboring u are satisfied
$$\prod_{v'\in\mathcal{N}_{\mathcal{U}^{(3)}\setminus v}(u)}\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\psi\left(u'\right)=0\middle|\begin{matrix}\psi\left(u\right)=b,\\M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\}.\tag{24}$$
3) The probability that all indicator parity-checks neighboring u are satisfied
$$\prod_{v'\in\mathcal{N}_{\mathcal{U}^{(1)}}(u)}\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\imath\left(u'\right)=0\middle|\begin{matrix}\psi\left(u\right)=b,\\M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\}.\tag{25}$$

Note that the first term is exactly $\mu^{(0)}_u(b)$ according to the initialization in Section III-C, and the second term is simply the product of residual check-to-variable messages from all neighbors of u excluding v, i.e., each term in the product in (24) equals $\mu^{(t)}_{v'\to u}(b)$. We now derive the third term for $b=1$.

$$\begin{aligned}&\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\imath\left(u'\right)=0\middle|\begin{matrix}\psi\left(u\right)=1,\\M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\}\\&=\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\imath\left(u'\right)=0\middle|\begin{matrix}\imath\left(u\right)=1,\\M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\}\\&=m^{(t)}_{v'\to u}(1).\end{aligned}\tag{26}$$

The first equality is because $\psi\left(u\right)=1$ implies $\imath\left(u\right)=1$, and the second equality is by the definition of check-to-variable messages in the indicator sub-graph. Similarly, for $b=0$ we have

$$\begin{aligned}&\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\imath\left(u'\right)=0\middle|\begin{matrix}\psi\left(u\right)=0,\\M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\}\\&=\frac{\rho_0}{\rho_0+\rho_1}\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\imath\left(u'\right)=0\middle|\begin{matrix}u=0,\\M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\}\\&+\frac{\rho_1}{\rho_0+\rho_1}\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\imath\left(u'\right)=0\middle|\begin{matrix}u=1,\\M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\}\\&=\frac{\rho_0}{\rho_0+\rho_1}\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\imath\left(u'\right)=0\middle|\begin{matrix}\imath\left(u\right)=0,\\M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\}\\&+\frac{\rho_1}{\rho_0+\rho_1}\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\imath\left(u'\right)=0\middle|\begin{matrix}\imath\left(u\right)=1,\\M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\}\\&=\frac{\rho_0 m^{(t)}_{v'\to u}(0)+\rho_1 m^{(t)}_{v'\to u}(1)}{\rho_0+\rho_1}.\end{aligned}\tag{27}$$

where the first equality is by marginalization over the two values of u corresponding to $\psi\left(u\right)=0$, the second equality is from applying the mapping $\imath\left(\cdot\right)$ to the values of u, and the last equality is by the definition of check-to-variable messages in the indicator sub-graph. To calculate the corresponding LLR, recall that $\mathcal{L}$ is the natural logarithm of the division of Eq. (27) and Eq. (26) (Definition 7). Substituting $\rho_0={}^1\!/_2, \rho_1={}^1\!/_4$, we get

$$\ln\left(\frac{1}{3}+\frac{2}{3}\exp\left(\mathcal{L}\left(m^{(t)}_{v'\to u}\right)\right)\right),\tag{28}$$

which equals $\mathcal{T}^{(3)}\left(\mathcal{L}\left(m^{(t)}_{v'\to u}\right)\right)$. We get (14) by applying Definition 7 on Eq. (23) and (24) as well.