Graph Codes for Dual-Parameter Barrier Channels

Authors
The Andrew and Erna Viterbi Department of Electrical and Computer Engineering
Technion - Israel Institute of Technology
Technion City, Haifa 3200003, Israel
Email: {abc@campus, def@ee}.technion.ac.il

***Abstract*—Decoding of codes over the barrier channel according to prior-art methods is either sub-optimal (sequential unique decoders) or computationally intensive (cooperative list decoders). This paper introduces a novel construction of graph-based barrier codes, which allows to perform cooperative decoding at low complexity using a modified message-passing algorithm. Simulation results of the suggested decoder demonstrates superior error rate performance over existing 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 consumption) 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]. Ternary representations were also considered for traditional CMOS logic [4] and wide-band communication systems [5].

The inherent asymmetry of a ternary representation implies that such devices would not admit the widely-adopted symmetric error models. We study in this work a ternary error model in which transitions between the highest level and the lowest level are either 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 $r = 0$ or $c = 0$ and $r \neq 0$. We refer to these two types of errors as *downward* and *upward* errors, respectively.

The barrier channel was addressed in [6] for $q = 0$ and in [3], [7] for $q = p$. The main contribution of [7] (and its extension [3]) is a construction method for ternary codes correcting t barrier errors, using a pair of binary Hamming-metric constituent codes. The decoder suggested in [3] for this construction decodes the two binary codes sequentially to produce the decoded ternary codeword. Enhanced decoders were developed in [8], [9], but at the cost of higher computational complexity. Since practical channels may not fall into one of the two previously studied special cases of the barrier channel, it is motivated to study the general case that allows asymmetry between the two barrier error types. Furthermore, the prior art decoding scheme is inherently sub-optimal due to its sequential nature which does not fully utilize the dependencies between the two constituent codes.

In this work, we develop a novel representation of ternary barrier codes using a bilayer Tanner graph. Based on the resulting graph code, we formulate the relations between the two constituent codes and harness them towards a new message-passing decoding algorithm. The uniqueness of this new decoder lies in its ability to consider information concerning both constituent codes *jointly* while decoding the ternary codeword. Based on exhaustive simulations, the newly suggested decoder is shown to significantly improve block error rates (BLER) over the dual-parameter barrier channel, compared to a prior-art decoder, for various code parameters and channel conditions.

The remainder of the paper is organized as follows. Section II contains preliminary results concerning prior art. In section III, we construct the barrier error correcting code as a graph code, devise a corresponding message-passing decoding algorithm and formulate the various messages passed during decoding. Section IV includes simulation results of our suggested decoder. The paper concludes in section V.

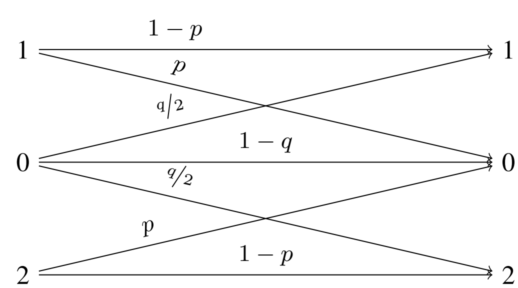

Fig. 1: The dual-parameter barrier channel for $Q = 3$.

II. Preliminaries: Code construction and decoding

Error correcting codes for various variants of the ternary Barrier channel were previously constructed in [3], [7], [8]. In this paper, we consider ternary codes constructed based on a decomposition to binary a indicator code Θ and a binary residual code Λ, defined as follows.

Definition 1 (Barrier code): Let $\Theta \subseteq \mathbb{Z}_2$ and $\Lambda \subseteq \mathbb{Z}_2$ be binary codes and let $n \in \mathbb{N}$. Then, every codeword $\mathbf{c} = (c_1, \ldots, c_n)$ in the ternary code $\mathcal{C} = \Theta \otimes \Lambda \subseteq \mathbb{Z}_3^n$ satisfies

1) There exists $\theta = (\theta_1, \ldots, \theta_n) \in \Theta$ such that $\imath(\mathbf{c}) = \theta$.
2) For this $\theta \in \Theta$, there exists $\lambda = (\lambda_1, \ldots, \lambda_n) \in \Lambda$ such that for every $j \in [n]$: $c_j = \lambda_j + 1$ if $\theta_j \neq 0$ and $c_j = \lambda_j = 0$ if $\theta_j = 0$.

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 $\mathbf{H}^{(\Theta)}$ and $\mathbf{H}^{(\Lambda)}$, respectively.

The decoder for $\mathcal{C}$ utilizes a mapping of the channel output to two constituent binary codewords, the indicator word and the residual word, based on the following mapping functions. Throughout the paper, we notate $\mathbb{Z}_Q \triangleq \{0, 1, Q-1\}$.

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

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

Definition 3: Let $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{Z}_Q^n$. The residual mapping of $\mathbf{x}$ is defined as $\psi(\mathbf{x}) \triangleq (\psi(x_1), \ldots, \psi(x_n))$ where

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

Given a channel output $\mathbf{y}$, prior art decoding algorithms usually employ a two step decoding scheme: First, a decoder for Θ is invoked for $\imath(\mathbf{y})$ to produce $\hat{\theta}$; then, a decoder for Λ is invoked for $\hat{\theta} \cdot \psi(\mathbf{y})$ to produce λ. Finally, the decoded constituent words are combined into the decoded codeword $\hat{\mathbf{c}} = \hat{\theta} + \hat{\lambda}$.

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

Although the indicator and residual codes are intertwined in the ternary codeword, the 2 step decoder only utilizes indicator error correction for easier residual erasure correction. Clearly this one-way information flow is less effective than a joint error correction incorporating the two codes for ternary symbol decoding. For example, whenever the residual decoder outputs a symbol with (binary) value $\hat{\lambda}_j = 1$, the corresponding ternary symbol is $\hat{c}_j = 2$, and consequently $\hat{\theta}_j = 1$. This information can assist the indicator decoder.

III. Graph decoding of Barrier codes

Any linear block code with parity-check matrix $\mathbf{H}$ 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 $\mathbf{H}_{i,j} = 1$. 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 and derive a corresponding message-passing decoding algorithm.

A. Bilayer Tanner graph

Although a Barrier code $\mathcal{C}$ is constructed using two constituent linear codes, Θ and Λ, $\mathcal{C}$ itself is not necessarily linear. Nevertheless, the unique structure of a ternary Barrier code allows to describe it using a bilayer graph.

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

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

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 $\mathbf{H}^{(\Theta)}$ and $\mathbf{H}^{(\Lambda)}$ 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 1. Define $\mathcal{U}^{(1)} = \{h_i\}_{i=1}^{n-k_I}$,$\mathcal{U}^{(2)} = \{w_i\}_{i=1}^n$ and $\mathcal{U}^{(3)} = \{u_i\}_{i=1}^{n-k_R}$ 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

$$\mathcal{E}^{(1)} = \left\{(w_i, h_j) : w_i \in \mathcal{U}^{(2)}, h_j \in \mathcal{U}^{(1)}, \mathbf{H}_{i,j}^{(\Theta)} \neq 0\right\}$$
$$\mathcal{E}^{(2)} = \left\{(w_r, u_j) : w_r \in \mathcal{U}^{(2)}, u_j \in \mathcal{U}^{(3)}, \mathbf{H}_{i,j}^{(\Lambda)} \neq 0\right\}$$

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

When the symbols of a codeword are assigned to the variable-nodes in a Tanner graph, the modulo-sum of the each node is 0. To satisfy a similar property in our bilayer graph, we apply an indicator mapping $\imath(w_i)$ on each $e \in \mathcal{E}^{(1)}$ connected to w_i (or residual mapping $\psi(w_i)$ on each $e \in \mathcal{E}^{(2)}$ connected to w_i, respectively) when the codeword *ternary* symbols are assigned to w_i. As we show in the following proposition, this operation corresponds to computing the j^{th} parity-check equation of the indicator code (or residual code, respectively).

Proposition 1: Let $\mathcal{C} = \Theta \otimes \Lambda$ be a Barrier code and let $\mathcal{G} = \left(\mathcal{U}^{(1)} \cup \mathcal{U}^{(2)} \cup \mathcal{U}^{(3)}, \mathcal{E}^{(1)} \cup \mathcal{E}^{(2)}\right)$ be its bilayer Tanner graph.

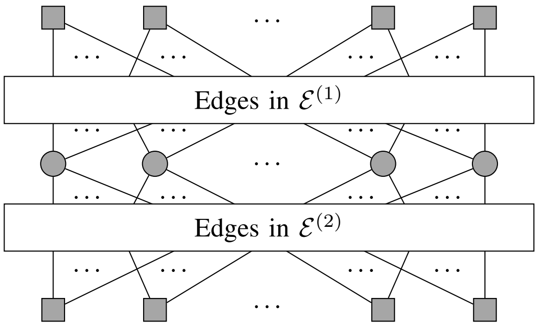

Fig. 2: An illustration of a bilayer Tanner graph

Then, the following properties are satisfied for any codeword $\mathbf{c} = (w_1, \ldots, w_n) \in \mathcal{C}$:

$$\sum_{w_j \in \mathcal{N}_{\mathcal{U}^{(2)}}(u)} \imath\left(w_j\right) = 0 \text{ for every } u \in \mathcal{U}^{(1)}, \tag{4}$$

$$\sum_{w_j \in \mathcal{N}_{\mathcal{U}^{(2)}}(u)} \psi\left(w_j\right) = 0 \text{ for every } u \in \mathcal{U}^{(3)}. \tag{5}$$

Proof: By definition, every codeword $c \in \mathcal{C}$ can be written as $c = \imath\left(c\right) + \psi\left(c\right)$, where $\imath\left(c\right) \in \Theta$ and $\psi\left(c\right) \in \Lambda$. Denote the indicator of the existence of an edge (u, v) as $\mathbb{I}(u, v)$. Then, for every $u_i \in \mathcal{U}^{(1)}$,

$$\begin{aligned} \sum_{w_j \in \mathcal{N}_{\mathcal{U}^{(2)}}(u_i)} \imath\left(w_j\right) &= \sum_{j=1}^{n} \mathbb{I}(w_j, u_i) \imath\left(w_j\right) \\ &= \sum_{j=1}^{n} \mathbf{H}^{(\Theta)}_{i,j} \imath\left(c_j\right) = \mathbf{H}^{(\Theta)} \imath\left(\mathbf{c}^{\top}\right) = 0, \end{aligned} \tag{6}$$

where the first and second transitions are by the definition of the Tanner graph $\mathcal{G}$, and the last transition is due to linearity of the indicator code Θ. According to similar arguments applied on the residual code, we get that for every $u_i \in \mathcal{U}^{(3)}$,

$$\begin{aligned} \sum_{w_j \in \mathcal{N}_{\mathcal{U}^{(2)}}(u_i)} \psi\left(w_j\right) &= \sum_{j=1}^{n} \mathbb{I}(w_j, u_i) \psi\left(w_j\right) \\ &= \sum_{j=1}^{n} \mathbf{H}^{(\Lambda)}_{i,j} \psi\left(c_j\right) = \mathbf{H}^{(\Lambda)} \psi\left(\mathbf{c}^{\top}\right) = 0. \end{aligned} \tag{7}$$

■

B. Message-passing joint decoding

Since decoding the ternary $\mathbf{c}$ directly is computationally hard, we present a decoding algorithm that utilizes the underlying binary constituent codes. Unlike prior art decoders, which first decode the indicator codeword and then the residual codeword, we suggest to interleave these two individual decoding processes towards a joint decoding scheme. Thanks to the bilayer Tanner graph representation, we were able to devise a decoding algorithm that allows to share intermediate decoding products between the indicator and residual codes while decoding $\mathcal{C}$.

To this end, we use a modification of the message-passing algorithm. Message-passing decoding employs an iterative procedure that transfers information (“messages”) across the edges of a graphical model. This information is used to compute the conditional probability of a message symbol w_i given an observed channel output y_i. In our suggested decoder, we apply a message-passing iteration (defined next) to perform partial decoding of the indicator (respectively, residual) code, and then move to decode the residual (respectively, indicator) code based on the most recent partial decoding results.

We devise a decoding sequence that can be employed for each of the constituent codes (indicator or residual). 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 $\mathcal{U}^{(2)}$) 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 an set of arbitrary nodes $\mathcal{U}$ to a target node v at iteration t as $M^{(t)}_{\mathcal{U} \to v} = \{m_{u' \to v}(\cdot)\}_{u' \in \mathcal{U}}$.

We first present the indicator decoding sequence, and then explain how it can be 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 $\mathbf{y} = (y_1, \ldots, y_n) \in \mathbb{Z}_3^n$, 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

$$m^{(t)}_{v_j \to u_i}(b) = \Pr\left\{ \sum_{u_k \in \mathcal{N}(v_j)} \imath\left(u_k\right) = 0 \middle| \begin{array}{l} \imath\left(u_i\right) = b, \\ M^{(t)}_{\mathcal{U}^{(2)} \setminus \{u_i\} \to v_j} \end{array} \right\}. \tag{8}$$

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

$$m^{(t)}_{u_i \to v_j}(b) = \Pr\left\{ \imath\left(u_i\right) = b \middle| \begin{array}{c} y_i,\, M^{(t)}_{\mathcal{U}^{(1)} \setminus v_j \to u_i}, \\ M^{(t)}_{\mathcal{U}^{(3)} \to u_i} \end{array} \right\}. \tag{9}$$

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 $\psi\left(\cdot\right)$ instead of $\imath\left(\cdot\right)$. We refer to these adapted terms as the *residual-modified* equations. To estimate the decoded indicator/residual bits, based on the previously calculated messages, for each node $u_i \in \mathcal{U}^{(2)}$, set

$$m^{(t)}_{u_i}(b) = \Pr\{f(u_i) = b \mid y_i, M_{\mathcal{U}^{(1)} \to u_i}, M_{\mathcal{U}^{(3)} \to u_i}\}, \tag{10}$$

where f is $\imath\left(\cdot\right)$ for indicator bits and $\psi\left(\cdot\right)$ for residual bits. 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, we present our iterative message-passing algorithm which uses these messages to decode $\mathcal{C}$. The proposed algorithm comprises of a maximum of T iterations, where in iteration executes either an indicator iteration or a residual iteration according to a predefined sequence $\mathbf{s} \in \{\text{‘Ind’}, \text{‘Res’}\}^T$. Before each such iteration, the decoded word is calculated and returned as output in case it is a codeword.

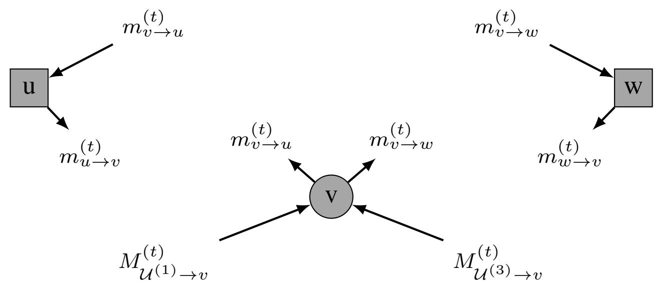

Fig. 3: Illustration of the messages accumulated and emanated from nodes in the bilayer Tanner graph, where $u \in \mathcal{U}^{(1)}, v \in \mathcal{U}^{(2)}$ and $w \in \mathcal{U}^{(3)}$

Algorithm 1 Joint message-passing decoding of Barrier codes

1: **Input:** Channel output $\mathbf{y} \in \mathbb{Z}_3^n$, bilayer Tanner graph $\mathcal{G}$, sequence $\mathbf{s}$
2: **Output:** Decoded codeword $\hat{c}$ or 'failure'
3: Initialize

$$\begin{aligned} m^{(0)}_{u_i}(b) &\leftarrow \Pr\{\imath\,(u_i) = b \mid y_i\} \\ \mu^{(0)}_{u_i}(b) &\leftarrow \Pr\{\psi\,(u_i) = b \mid y_i\} \end{aligned} \tag{11}$$

4: **for** $t = 1, \ldots, T$ **do**
5: Update $\Pr(w_i | y_i, M^{(t)}_{\mathcal{U}^{(1)}\to w_i}, M^{(t)}_{\mathcal{U}^{(3)}\to w_i})$ for $w_i = 0, 1, 2$ and $i = 1, \ldots, n$. Then, calculate $\hat{\mathbf{c}} = (\hat{c}_1, \ldots, \hat{c}_n)$ where

$$\hat{c}_i = \arg\max \Pr(w_i | y_i, M^{(t)}_{\mathcal{U}^{(1)}\to w_i}, M^{(t)}_{\mathcal{U}^{(3)}\to w_i}) \tag{12}$$

6: **if** $\hat{\mathbf{c}}$ is a codeword in $\mathcal{C}$ **then**
7: **return** $\hat{\mathbf{c}}$
8: **end if**
9: **if** s_t equals 'Ind' **then**
10: Perform indicator iteration: Apply Eq. (8), then apply Eq. (9)
11: **else**
12: Perform residual iteration: Apply residual-modified Eq. (8), then apply residual-modified Eq. (9)
13: **end if**
14: **end for**
15: **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. 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 $m(\cdot)^{(t)}$ a message on the graph. Then the LLR is defined as $L(m^{(t)}) = \ln\left(\frac{m^{(t)}(0)}{m^{(t)}(1)}\right)$.

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

Proposition 2: 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,

$$\mathcal{L}\left(m^{(0)}_u\right) = \begin{cases} \ln\left(\frac{1-q}{p}\right), & \text{if } y = 0 \\ \ln\left(\frac{q}{1-p}\right), & \text{if } y = 1 \\ \ln\left(\frac{q}{1-p}\right), & \text{if } y = 2 \end{cases}, \tag{13}$$

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

Proof: The initial indicator message is defined as $m^{(0)}_u = \ln\left(\frac{m^{(0)}_u(0)}{m^{(0)}_u(1)}\right)$. Denote the prior probability $\rho_{u'} \triangleq \Pr\{u = u'\}$ for $u' = 0, 1, 2$. Hence,

$$\begin{aligned} \mathcal{L}\left(m^{(0)}_u\right) &= \ln\left(\frac{\Pr\{\imath\,(u) = 0 \mid y\}}{\Pr\{\imath\,(u) = 1 \mid y\}}\right) \\ &= \ln\left(\frac{\Pr\{y \mid u = 0\}\rho_0}{\Pr\{y \mid u = 1\}\rho_1 + \Pr\{y \mid u = 2\}\rho_2}\right) \\ &= \begin{cases} \ln\left(\frac{1-q}{p}\right), & \text{if } y = 0 \\ \ln\left(\frac{q}{1-p}\right), & \text{if } y = 1 \\ \ln\left(\frac{q}{1-p}\right), & \text{if } y = 2 \end{cases}, \end{aligned} \tag{14}$$

where the last transition is according to the Barrier channel definition. Similarly, the initial residual message is

$$\begin{aligned} \mathcal{L}\left(\mu^{(0)}_u\right) &= \ln\left(\frac{\Pr\{\psi\,(u) = 0 \mid y\}}{\Pr\{\psi\,(u) = 1 \mid y\}}\right) \\ &= \ln\left(\frac{\Pr\{y \mid u = 0\}\rho_0 + \Pr\{y \mid u = 1\}\rho_1}{\Pr\{y \mid u = 2\}\rho_2}\right) \\ &= \begin{cases} \ln\left(1 + 2\frac{1-q}{p}\right), & \text{if } y = 0 \\ \infty, & \text{if } y = 1 \\ \ln\left(\frac{q}{1-p}\right), & \text{if } y = 2 \end{cases}. \end{aligned} \tag{15}$$

■

At each iteration t, the messages are sent from check nodes to variable nodes, and then vice-versa. Since each bipartite sub-graph 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 [10]. 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,

$$\mathcal{L}\left(m^{(t)}_{v\to u}\right) = \left(\prod_{u' \in \mathcal{N}_{u,v}} \alpha^{(t-1)}_{u',v}\right) \phi\left(\sum_{u' \in \mathcal{N}_{u,v}} \phi\left(\beta^{(t-1)}_{u',v}\right)\right), \tag{16}$$

where $\phi(x) \triangleq \ln\left(\frac{e^x+1}{e^x-1}\right)$, $\alpha^{(\tau)}_{u',v} = \text{sign}\left(\mathcal{L}\left(m^{(\tau)}_{u'\to v}\right)\right)$ and $\beta^{(\tau)}_{u',v} = \left|\mathcal{L}\left(m^{(\tau)}_{u'\to v}\right)\right|$. Similarly, the residual check to variable messages can be calculated as follows. For a residual check node $v \in \mathcal{U}^{(3)}$ and a variable node $u \in \mathcal{U}^{(2)}$, we calculate $\mathcal{L}\left(\mu^{(t)}_{v\to u}\right)$ by applying the following modifications to Eq. 16: (1) $\alpha^{(\tau)}_{u',v} = \text{sign}\left(\mathcal{L}\left(\mu^{(\tau)}_{u'\to v}\right)\right)$ (2) $\beta^{(\tau)}_{u',v} \triangleq \left|\mathcal{L}\left(\mu^{(\tau)}_{u'\to v}\right)\right|$.

The variable to check messages require a more complicated derivation, 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 3 and then the residual variable to check messages in Proposition 4.

Proposition 3 (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,

$$\begin{aligned}\mathcal{L}(m^{(t)}_{u\to v}) =& \mathcal{L}(m^{(0)}_u) + \sum_{v' \in \mathcal{N}_{\mathcal{U}^{(1)}\setminus v}(u)} \mathcal{L}(m^{(t)}_{v'\to u}) \\ &+ \sum_{v' \in \mathcal{N}_{\mathcal{U}^{(3)}}(u)} \mathcal{T}^{(1)}\left(\mathcal{L}\left(\mu^{(t)}_{v'\to u}\right)\right).\end{aligned}$$

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 similar fashion in the following proposition.

Proposition 4 (variable to residual check): Let $u \in \mathcal{U}^{(2)}$ be a variable node and let $v \in \mathcal{U}^{(3)}$ be an residual parity-check node. Then,

$$\begin{aligned}\mathcal{L}(\mu^{(t)}_{u\to v}) =& \mathcal{L}\left(\mu^{(0)}_u\right) + \sum_{v' \in \mathcal{N}_{\mathcal{U}^{(3)}\setminus v}(u)} \mathcal{L}\left(\mu^{(t)}_{v'\to u}\right) \\ &+ \sum_{v' \in \mathcal{N}_{\mathcal{U}^{(1)}}(u)} \mathcal{T}^{(3)}\left(\mathcal{L}\left(\mu^{(t)}_{v'\to u}\right)\right).\end{aligned}$$

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 message sent from a residual check node to u into the proper contribution to a message sent from u to an indicator check node. For example, $\mathcal{T}^{(1)}(-\infty)$ implies that $\psi(u) = 1$, thus $\imath(u) = 1$, and hence the indicator bit's LLR also tends to $-\infty$. Conversely, $\mathcal{T}^{(1)}(\infty)$ implies $\psi(u) = 0$, whose LLR contribution to the indicator code reflects the prior indicator bit distribution. A similar logic applies to $\mathcal{T}^{(3)}$, which translates information in the reverse direction, i.e., indicator LLRs to residual LLRs. Both functions are illustrated in Fig. 4.

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 $\Pr\left\{u|y, M^{(t)}_{\mathcal{U}^{(1)}\to u}, M^{(t)}_{\mathcal{U}^{(3)}\to u}\right\}$. Writing the expression for the

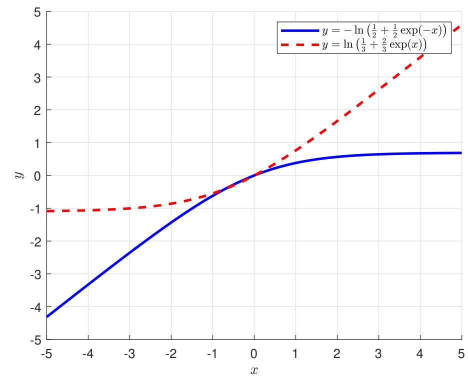

Fig. 4: Illustration of LLR functions assembling the variable to check messages of the joint decoder from Propositions 3 and 4

probability of $u = 0$ explicitly, based on indicator messages, leads to

$$\Pr\left\{u = 0|y, M^{(t)}_{\mathcal{U}^{(1)}\to u}, M^{(t)}_{\mathcal{U}^{(3)}\to u}\right\} = \frac{1}{1 + \exp\left(-\mathcal{L}\left(m^{(t)}_u\right)\right)}, \tag{17}$$

where $m^{(t)}_u$ is defined in (10) (with $f(\cdot) = \imath(\cdot)$). Similarly, the corresponding probability for $u = 2$ is given by

$$\Pr\left\{u = 2|y, M^{(t)}_{\mathcal{U}^{(1)}\to u}, M^{(t)}_{\mathcal{U}^{(3)}\to u}\right\} = \frac{1}{1 + \exp\left(\mathcal{L}\left(\mu^{(t)}_u\right)\right)}, \tag{18}$$

where $\mu^{(t)}_u$ is defined in (10) (with $f(\cdot) = \psi(\cdot)$). 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 was done using Low-Density Parity-Check (LDPC) codes. To construct the Ternary Barrier code, we used 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 1), we pass them through the (p, q) dual-parameter Barrier channel $W_3(p, q)$. The erroneous word on the channel's output is then passed as input to the simulated decoders.

We simulated our suggested 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, as suggested in prior-work, which employs the following sequence: (1) Perform 30 indicator decoding iterations on $\imath(\mathbf{y})$ to obtain a decoded indicator codeword $\hat{\theta}$; (2) Generate an indicator-induced residual word $\mathbf{y}' = \hat{\theta}\cdot\psi(\mathbf{y})$; and (3) Employ an erasure decoder on $\mathbf{y}'$ where locations in which $y_i \neq 0$ and $\hat{\theta}_i = 0$ are marked as erasures.

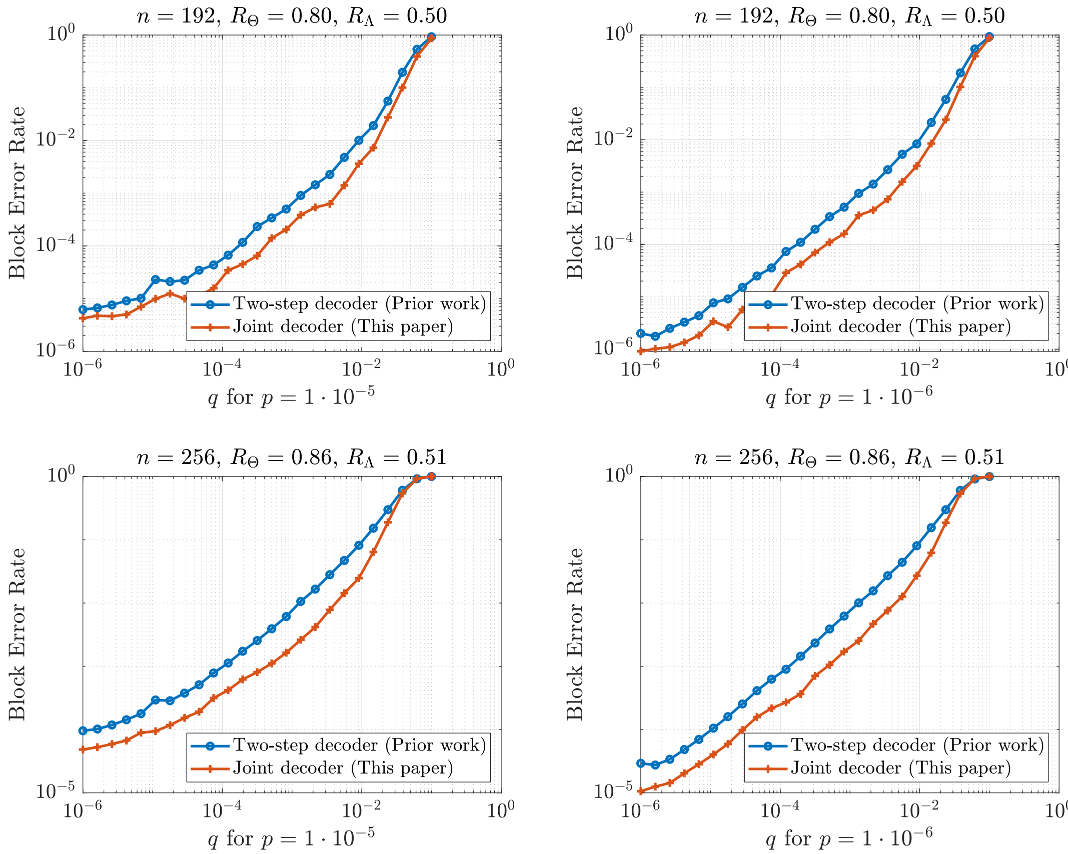

Fig. 5: Block error rates (BLER) of the suggested joint decoder and a baseline prior-work decoder simulated over $W_3(p, q)$ for $p = 10^{-5}$ (left), $p = 10^{-6}$ (right) and code lengths $n = 192$ (up), $n = 256$ (down)

In our simulations, we focused on non-symmetric Barrier channels, in which the downward error probability p is typically smaller than the upward error probability q. We also set the residual rate to 0.5 while the indicator rate was set to higher values (0.8 or 0.86). In such settings, the residual code is both more robust and experiences less errors, hence it may assist the indicator decoding. The decoding sequence $\mathbf{s}$ of the joint decoder was arranged in sub-sequences of $s^{(1)}$ indicator iterations followed by $s^{(2)}$ residual iterations, i.e.,

$$\mathbf{s} = \left[\underbrace{\text{`Ind'}, \ldots, \text{`Ind'}}_{t_i \text{ times}}, \underbrace{\text{`Res'}, \ldots, \text{`Res'}}_{t_r \text{ times}}, \ldots \right].$$

Specifically, we tested several parameter constellations 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 $\mathbf{s}$, 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), i.e., the fraction of transmitted codewords whose corresponding decoded codewords contained one or more symbol errors, over large numbers of codeword and channel realizations. Figure 5 presents the results of such experiment for several parameter sets: codeword length of $n = 128$ and $n = 256$; downward error probability p of 10^{-5} and 10^{-6}; and q ranging between 10^{-6} and 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. Moreover, the gap from baseline performance seems to be greater for $n = 256$ compared to $n = 128$, indicating that larger block lengths may further enjoy the advantage of joint decoding.

V. Conclusion

This paper considers graph codes for error correction over the dual-parameter barrier channel. Based on a novel construction using bilayer Tanner graphs, we devise an innovative decoding algorithm that is specifically designed for the unique structure of barrier codes. Throughout the decoding process, the suggested 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 research of adaptive sequence optimization during decoding, rather than the fixed sequence used in our simulations.

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, 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.

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

[5] 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.

[6] 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.

[7] 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.

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

APPENDIX A
PROOF OF PROPOSITION 3

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

1) The (normalized) posterior probability of u, denoted $\eta_{u,v}^{(\mu)}(t)\Pr\{\imath(u)=b\mid y\}$.
2) The probability that all indicator parity-check equations, whose corresponding node is denoted v, involving u are satisfied

$$\prod_{v'\in\mathcal{N}_{\mathcal{U}^{(1)}\setminus v}(u)}\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\imath(u')=0\middle|\begin{matrix}\imath(u)=b,\\ M_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}^{(t)}\end{matrix}\right\} \tag{19}$$

3) The probability that every residual parity-check equation, whose corresponding node is denoted v, involving u is satisfied

$$\prod_{v'\in\mathcal{N}_{\mathcal{U}^{(3)}}(u)}\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\psi(u')=0\middle|\begin{matrix}\imath(u)=b,\\ M_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}^{(t)}\end{matrix}\right\} \tag{20}$$

Note that the first term is exactly $m_u^{(0)}(b)$, and the second term is simply the product of indicator check to variable messages from all neighbors of u excluding v, i.e., $m_{v'\to u}^{(t)}(b)$ where $v'\in\mathcal{N}_{\mathcal{U}^{(1)}\setminus v}(u)$. We now derive the third term.

$$\begin{aligned}
&\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\psi(u')=0\middle|\begin{matrix}\imath(u)=0,\\ M_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}^{(t)}\end{matrix}\right\}\\
&=\Pr\left\{\sum_{u'\in\mathcal{N}(v')\setminus u}\psi(u')=\psi(u)\middle|\begin{matrix}\imath(u)=0,\\ M_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}^{(t)}\end{matrix}\right\}\\
&=\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\psi(u')=\psi(u)\middle|\begin{matrix}\psi(u)=0,\\ M_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}^{(t)}\end{matrix}\right\}\\
&=\mu_{v'\to u}^{(t)}(0).
\end{aligned} \tag{21}$$

The second transition is true since conditioning the probability of the sum equaling to $\psi(z)$ by $\imath(z)=0$ is the same as conditioning it by $\psi(z)=0$; and the last transition is by definition. Similarly, for $b=1$ we have

$$\begin{aligned}
&\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\psi(u')=0\middle|\begin{matrix}\imath(u)=1,\\ M_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}^{(t)}\end{matrix}\right\}\\
&=\Pr\left\{\sum_{u'\in\mathcal{N}(v')\setminus u}\psi(u')=\psi(u)\middle|\begin{matrix}\imath(u)=1,\\ M_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}^{(t)}\end{matrix}\right\}\\
&=\frac{\rho_1}{\rho_1+\rho_2}\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\psi(u')=\psi(u)\middle|\begin{matrix}u=1,\\ M_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}^{(t)}\end{matrix}\right\}\\
&+\frac{\rho_2}{\rho_1+\rho_2}\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\psi(u')=\psi(u)\middle|\begin{matrix}u=2,\\ M_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}^{(t)}\end{matrix}\right\}\\
&=\frac{\rho_1}{\rho_1+\rho_2}\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\psi(u')=\psi(u)\middle|\begin{matrix}\psi(u)=0,\\ M_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}^{(t)}\end{matrix}\right\}\\
&+\frac{\rho_2}{\rho_1+\rho_2}\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\psi(u')=\psi(u)\middle|\begin{matrix}\psi(u)=1,\\ M_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}^{(t)}\end{matrix}\right\}\\
&=\frac{\rho_1\mu_{v'\to u}^{(t)}(0)+\rho_2\mu_{v'\to u}^{(t)}(1)}{\rho_1+\rho_2}.
\end{aligned} \tag{22}$$

We get (17) by setting $\rho_1=\rho_2={}^1/_4$ and recalling the definition of $\mathcal{L}$.

APPENDIX B
PROOF OF PROPOSITION 4

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

1) The (normalized) posterior probability of u, denoted $\eta_{u,v}^{(\mu)}(t)\Pr\{\psi(u)=b\mid y\}$.
2) The probability that all residual parity-check equations, whose corresponding node is denoted v, involving u are satisfied

$$\prod_{v'\in\mathcal{N}_{\mathcal{U}^{(3)}\setminus v}(u)}\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\psi(u')=0\middle|\begin{matrix}\psi(u)=b,\\ M_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}^{(t)}\end{matrix}\right\} \tag{23}$$

3) The probability that every indicator parity-check equation, whose corresponding node is denoted v, involving u is satisfied

$$\prod_{v'\in\mathcal{N}_{\mathcal{U}^{(1)}}(u)}\Pr\left\{\sum_{u'\in\mathcal{N}(v')}\imath(u')=0\middle|\begin{matrix}\psi(u)=b,\\ M_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}^{(t)}\end{matrix}\right\} \tag{24}$$

Note that the first term is exactly $\mu_u^{(0)}(b)$, and the second term is simply the product of residual check to variable messages from all neighbors of u excluding v, i.e., $\mu^{(t)}_{v'\to u}(b)$ where $v' \in \mathcal{N}_{\mathcal{U}^{(3)}\setminus v}(u)$. We now derive the third term for $b = 1$.

$$\begin{aligned}
&\Pr\left\{\sum_{u'\in\mathcal{N}(v')} \imath\left(u'\right) = 0 \middle| \begin{matrix}\psi\left(u\right) = 1, \\ M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\} \\
&\quad = \Pr\left\{\sum_{u'\in\mathcal{N}(v')\setminus u} \imath\left(u'\right) = \imath\left(u\right) \middle| \begin{matrix}\psi\left(u\right) = 1, \\ M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\} \\
&\quad = \Pr\left\{\sum_{u'\in\mathcal{N}(v')} \imath\left(u'\right) = \imath\left(u\right) \middle| \begin{matrix}\imath\left(u\right) = 1, \\ M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\} \\
&\quad = m^{(t)}_{v'\to u}(1).
\end{aligned} \tag{25}$$

The second transition is true since conditioning the probability of the sum equaling to $\psi\left(z\right)$ by $\imath\left(z\right) = 0$ is the same as conditioning it by $\psi\left(z\right) = 1$; and the last transition is by definition. Similarly, for $b = 0$ we have

$$\begin{aligned}
&\Pr\left\{\sum_{u'\in\mathcal{N}(v')} \imath\left(u'\right) = 0 \middle| \begin{matrix}\psi\left(u\right) = 0, \\ M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\} \\
&\quad = \Pr\left\{\sum_{u'\in\mathcal{N}(v')\setminus u} \imath\left(u'\right) = \imath\left(u\right) \middle| \begin{matrix}\psi\left(u\right) = 0, \\ M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\} \\
&\quad = \frac{\rho_0}{\rho_0+\rho_1}\Pr\left\{\sum_{u'\in\mathcal{N}(v')} \imath\left(u'\right) = \imath\left(u\right) \middle| \begin{matrix}u = 0, \\ M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\} \\
&\quad + \frac{\rho_1}{\rho_0+\rho_1}\Pr\left\{\sum_{u'\in\mathcal{N}(v')} \imath\left(u'\right) = \imath\left(u\right) \middle| \begin{matrix}u = 1, \\ M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\} \\
&\quad = \frac{\rho_0}{\rho_0+\rho_1}\Pr\left\{\sum_{u'\in\mathcal{N}(v')} \imath\left(u'\right) = \imath\left(u\right) \middle| \begin{matrix}\imath\left(u\right) = 0, \\ M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\} \\
&\quad + \frac{\rho_1}{\rho_0+\rho_1}\Pr\left\{\sum_{u'\in\mathcal{N}(v')} \imath\left(u'\right) = \imath\left(u\right) \middle| \begin{matrix}\imath\left(u\right) = 1, \\ M^{(t)}_{\mathcal{U}^{(2)}\setminus\{u\}\to v'}\end{matrix}\right\} \\
&\quad = \frac{\rho_0 m^{(t)}_{v'\to u}(0) + \rho_1 \mu^{(t)}_{v'\to u}(1)}{\rho_0+\rho_1}.
\end{aligned} \tag{26}$$

We get (17) by setting $\rho_0 = {}^1\!/_2, \rho_1 = {}^1\!/_4$ and recalling the definition of $\mathcal{L}$.