In-Memory Bit Error Rate Estimation Using Syndromes of LDPC Codes

Yotam Gershon Yuval Cassuto
The Viterbi Faculty of Electrical and Computer Engineering, Technion - Israel Institute of Technology
Email: {yotamgr@campus, ycassuto@ee}.technion.ac.il

***Abstract*—Modern AI systems entail steep energy costs due to massive-scale computations and data transfers; offloading parts of the computations to be performed in-memory holds great potential for reducing both. This paper studies a new architecture proposed for reliable in-memory computations. Its main component is a coding scheme that is designed for both in-memory error-rate estimation/detection and outside-of-memory error correction. Estimation and/or detection are used to decide when the error rate exceeds the tolerance of the computation, at which point error correction is invoked. The coding scheme is based on a nested bilayer LDPC construction, where in particular, the first layer comprises degree-1 variable nodes guaranteeing accurate bit-error rate (BER) estimation and detection. Towards that, we derive a closed-form maximum-likelihood BER estimator for irregular codes, and a gapped hypothesis testing framework for deciding when to decode given some prescribed error-rate tolerance. The performance analysis of the derived estimator includes a closed-form mean-squared-error expression with explicit dependence on the check-degree distribution. For the hypothesis testing the analysis shows the dependence of detection performance on the same degree distribution. Both results reveal an advantage of check-regular codes that minimize dominant error terms among codes with a given average check degree.**

I. Introduction

In-memory computing is a paradigm aiming to address the emerging bottlenecks of data-transfer rates and power to and from the memory [1], termed as the memory wall [2]. Based upon introducing processing capabilities into the memory itself, this approach gains increasing momentum in the last decade, mainly due to the ever increasing demand for efficient artificial intelligence (AI) computing platforms. AI systems perform massive computations on vast data, hence porting key AI primitives to in-memory accelerators may improve performance and reduce power significantly. Many architectures have been proposed to enable processing without transferring data outside the memory, ranging from implementing logic gates within traditional charge-based memories (mainly SRAM [3], [4] and DRAM [5]), to utilizing inherent properties of resistance-based memories for quantized analog computations (mainly RRAM [6] and MRAM [7]).

However, these highly efficient architectures also present challenges, one of them being maintaining reliability. On one hand, data integrity is challenged and degraded by the dynamic nature of the data and frequent read/write accesses, and on the other hand, error mitigation becomes challenging due to resource and latency constraints limiting the ability to employ powerful error-correction coding (ECC). The traditional way in which ECC is used for memory reliability is encoding before each write and decoding after each read [8]. This method is problematic for in-memory computing because of the fine access granularity and limited computing resources that challenge this approach. There have been proposals to implement ECC in memory (e.g. [9]), however, modern coding techniques, such as LDPC [10], [11] and polar [12], [13] codes, are difficult to decode with in-memory logic, while more traditional, easier-to-decode coding schemes are costly in redundancy.

That said, in-memory computing architectures can usually allow for a certain fraction of errors in the stored data, since many modern computing applications, especially in the fields of AI and machine learning [14] and approximate computing [15], can tolerate some errors throughout computations. This allows to divide the reliability maintenance between: 1) simple error-control tasks performed in-memory, and 2) powerful error-correction tasks performed outside the memory only infrequently, when higher reliability is needed. Under this setting, the paper studies two error-control tasks to be performed in-memory: *bit-error rate (BER) estimation*, and *BER-threshold crossing detection*. Both tasks are performed using the sparse check equations of LDPC codes, in a complementary way to the common use of these codes as a powerful error-correction scheme.

In this paper we study LDPC codes that are particularly designed for these two error-control tasks. More specifically, we design degree distributions (DD) that enable accurate estimation and threshold detection of BER from the syndrome weight (that can be easily evaluated in-memory without decoding). We use a special class of codes that have degree-1 variable nodes, which we call *non-intersecting variable nodes (NIV)*. NIV codes enjoy the useful property of statistically independent check equations. For error correction NIV codes are not useful in themselves, but we show how to concatenate them to get any desired DD for error correction. This allows to implement the codes in a nested architecture: a subset of parity checks is used for estimation/detection (or both), while the full set is used for error correction. The paper studies the DD's performance in both estimation and detection, while correction is covered by the vast literature on LDPC analysis and design. In a practical use of these codes in an in-memory architecture, BER estimation provides a fine-grained evaluation of the reliability, while BER-threshold detection is a coarser reliability measure that requires specifying a tolerable level of BER. Either of the

This work was supported in part by NSF-BSF grant 2023627. Part of this work was presented at the 2025 IEEE International Symposium on Information Theory (ISIT).

two enables the split-maintenance approach suggested above: only when the (periodically measured) BER level is declared intolerable, the data is transferred to a processing unit that decodes the full LDPC code; then it re-writes the corrected codeword to memory (and repeating indefinitely).

A. Related Work

BER estimation from syndromes was studied in [16], where a closed-form maximum likelihood (ML) estimator was derived for check-regular codes, and a mean squared error (MSE) analysis was performed using the known Cramer-Rao lower bound. A general form for check-irregular codes was provided, without a closed-form expression due to its complexity. All the results in [16] are derived under the assumption that syndrome values (representing satisfaction of parity-check constraints) are independent between different parity checks. In [17] this assumption is relaxed through an explicit analysis of the syndrome weight variance, taking into account the statistical dependence. Then, a normal distribution approximation, described by the expectation (already known from [16]) and this exact variance, is used to more accurately capture the syndrome weight distribution.

B. Paper Organization and Contributions

In Section III we explore the relation between the error weight (and equivalently, error rate) and the syndrome weight, which are the basis for BER estimation from syndromes, from the perspective of both variable and check nodes. In Section IV we describe the memory reliability architecture based on in-memory estimation+detection and out-of-memory correction. The heart of the architecture is a nested LDPC coding scheme with NIV codes for estimation. We examine the properties of such NIV codes, show how to use them for estimation while augmenting them with additional parity checks for powerful correction capability in the out-of-memory code. We specify the correction-code construction through the usual LDPC degree-distribution pair, but appropriately constrained in a bi-layer structure to guarantee the NIV property of the estimation code. Then, in Section V we pursue BER estimation by deriving ML estimators. We provide an estimator for the general check-irregular case, shown to generalize the previously known estimator for check-regular codes. We continue to analyze the estimator's performance, focusing on a closed-form expression for the MSE, with an explicit dependence on the check DD. We use this explicit dependence to understand the main aspects of DD's influence on the MSE, and show that, in the regime of small BER, the dominant terms of the MSE are monotone increasing with the DD's variance, and are thus minimized by check-regular codes for any given average check degree. In Section VI we move to BER-threshold detection using a gapped hypothesis-testing framework (has a "don't care" gap between hypotheses). We derive the type-I and type-II detection errors as a function of the code's DD, and then use approximate surrogates to lower bound the combined detection performance. This analysis also provides an optimization tool for setting the DD to maximize performance. Finally, in Section VII we present numerical simulations to both illustrate and support the findings in previous sections. Along with the introduction of an in-memory error control architecture, the main theoretical contributions of this work are as follows:

1) Generalizing ML estimation to irregular codes (based on an approximated likelihood function),
2) Providing a closed-from expression for the DD-dependent estimation MSE, along with a simplified expression for its dominant terms, and showing that check-regular codes minimize these dominant terms,
3) Introducing a gapped hypothesis-testing framework (with a "don't care" region) for deciding when to decode, using the gap for analyzing the DD-dependent detection performance, and showing the optimality of check-regular codes with respect to a surrogate lower bound on the detection performance.

II. Preliminaries

A. Notations

We denote by $\mathbf{C}^n(\Lambda,\Omega)$ an ensemble of irregular LDPC codes of length n and DD $\Lambda(x) \triangleq \sum_{i=1}^{d_v}\Lambda_i x^i$ and $\Omega(x) \triangleq \sum_{i=1}^{d_c}\Omega_i x^i$, with Ω_i,Λ_i describing the relative fraction of variable and check nodes of degree i, respectively. We assume $\Omega_1 = 0$ to prevent variable nodes fixed to 0. For check-regular (resp. variable-regular) codes with $\Omega(x) = x^{d_c}$ (resp. $\Lambda(x) = x^{d_v}$), we denote the ensemble explicitly by $\mathbf{C}^n(\Lambda, x^{d_c})$ (resp. $\mathbf{C}^n(x^{d_v},\Omega)$). The class of check-regular codes, sometimes referred to as right-regular, is studied extensively in the literature (e.g. [18]–[21]), and will be of a specific interest throughout the paper. We denote $m \triangleq n\Lambda^{'}(1)/\Omega^{'}(1)$ as the number of check nodes, where $'$ represents derivative with respect to x. The parity-check matrix (PCM) of size $m\times n$ of $\mathcal{C}\in\mathbf{C}^n(\Lambda,\Omega)$ is denoted $\mathcal{H}$. For any word $\boldsymbol{x}\in\{0,1\}^n$ the *syndrome* is defined by $\boldsymbol{s_x} \triangleq \mathcal{H}\boldsymbol{x}^T$ (with $\boldsymbol{s_x} = \mathbf{0}$ for codewords). The well-known Hamming weight and Hamming distance will be denoted by $w_H(\cdot)$ and $d_H(\cdot,\cdot)$, respectively. The Tanner graph [22] corresponding to $\mathcal{H}$ is denoted by $\mathcal{G}_{\mathcal{H}} = (\mathcal{V}\cup\mathcal{U},\mathcal{E})$ with $\mathcal{V} = \{v_1,\ldots,v_n\}$,$\mathcal{U} = \{u_1,\ldots,u_m\}$ denoting the sets of variable and check nodes, and $\mathcal{E} = \{(v_i,u_j) : \mathcal{H}_{ji} = 1\}$ denoting the edges. The degree of a node $v\in\mathcal{V}$ or $u\in\mathcal{U}$ is denoted by $\deg(v)$ or $\deg(u)$, respectively. For a natural number N we denote $[N] \triangleq \{1,2,\ldots,N\}$. $\mathrm{P}(Z=z),\mathbb{E}[Z]$ denote the probability distribution and expectation of a random variable (RV) Z, respectively. $\mathcal{O}(\cdot)$ will denote the big-O notation [23]. Finally, We define the cumulative Binomial probability of an odd number of successes by

$$p_{\mathrm{o}}(n,p) \triangleq \sum_{\substack{k\in\{0,\ldots,n\}\\ k \text{ odd}}}\binom{n}{k}p^k(1-p)^{n-k} = \frac{1-(1-2p)^n}{2}.$$

B. Memory Channel Model

For some word $\boldsymbol{c}\in\{0,1\}^n$ stored in memory, we assume the occurrence of bit-flips, replacing $\boldsymbol{c}$ with $\boldsymbol{c}'$, the result of passing $\boldsymbol{c}$ through a memoryless binary symmetric channel (BSC) with crossover probability $p\in[0,0.5]$. Let $\boldsymbol{e} \triangleq \boldsymbol{c}'\oplus\boldsymbol{c}\in\{0,1\}^n$, with $\oplus$ denoting an element-wise XOR operation, be the error vector. Let $w_{\boldsymbol{e}} \triangleq w_H(\boldsymbol{e}) = d_H(\boldsymbol{c},\boldsymbol{c}')$ be the error weight, which is a Binomial random variable with parameter p. The BER is defined by $w_{\boldsymbol{e}}/n$. Unless stated otherwise,

$\boldsymbol{s} \triangleq \boldsymbol{s}_{\boldsymbol{c}'} = \boldsymbol{s}_{\boldsymbol{e}}$ will denote the measured syndrome of the noisy $\boldsymbol{c}'$. The syndrome weight $w_{\boldsymbol{s}} \triangleq w_H(\boldsymbol{s})$ will be the main statistic for BER estimation (i.e., estimating $w_{\boldsymbol{e}}$ from $w_{\boldsymbol{s}}$). With a slight abuse of terminology, we will interchangeably discuss estimating the channel parameter p and the empirical error weight $w_{\boldsymbol{e}}$.

III. Relations Between Error Weight and Syndrome Weight

In this section we first formulate the relations between the error weight and the weight of the corresponding syndrome with respect to an LDPC code, which will form the basis for estimating the former from the latter.

A. Combinatorial Variable-Node Perspective Analysis

For a given error vector $\boldsymbol{e} = (e_1, \ldots, e_n)$, it is clear that only the subset $\mathcal{V}_{\boldsymbol{e}} = \{v_i \in \mathcal{V} : e_i = 1\}$, that is, nodes with erroneous value, affect the syndrome $\boldsymbol{s}$. Thus for ease of notation, we describe $\mathcal{V}_{\boldsymbol{e}}$ using consecutive indices $\{\bar{v}_1, \ldots, \bar{v}_{w_{\boldsymbol{e}}}\}$. We write $([w_{\boldsymbol{e}}]^l)$ to denote all the subsets $(i_1, \ldots, i_l)$ of $[w_{\boldsymbol{e}}]$ with l elements such that $i_1 < i_2 < \cdots < i_l$, and for each $\bar{v} \in \mathcal{V}_{\boldsymbol{e}}$, we denote by $\mathcal{U}(\bar{v})$ the subset of check nodes that are connected to v. The following theorem describes $w_{\boldsymbol{s}}$ as a function of $w_{\boldsymbol{e}}$ and $\mathcal{U}(\mathcal{V}_{\boldsymbol{e}})$.

Theorem 1. *Given an error vector $\boldsymbol{e}$ of weight $w_{\boldsymbol{e}}$,*

$$\begin{aligned} w_{\boldsymbol{s}} &= \sum_{l=1}^{w_{\boldsymbol{e}}} (-2)^{l-1} \sum_{([w_{\boldsymbol{e}}]^l)} \left| \bigcap_{j=1}^{l} \mathcal{U}(\bar{v}_{i_j}) \right| \\ &= \sum_{i=1}^{w_{\boldsymbol{e}}} \deg(\bar{v}_i) - 2 \sum_{1 \le i < j \le w_{\boldsymbol{e}}} |\mathcal{U}(\bar{v}_i) \cap \mathcal{U}(\bar{v}_j)| + \\ &\quad + 4 \sum_{1 \le i < j < k \le w_{\boldsymbol{e}}} |\mathcal{U}(\bar{v}_i) \cap \mathcal{U}(\bar{v}_j) \cap \mathcal{U}(\bar{v}_k)| - \ldots . \end{aligned} \tag{1}$$

Proof: See Appendix A-A. ∎

Theorem 1 implies several facts on the parity of $w_{\boldsymbol{s}}$. We notice that all terms but the first in Eq. (1) have even coefficients. Hence, the parity of $w_{\boldsymbol{s}}$ is determined solely by the first term, $\sum_{i=1}^{w_{\boldsymbol{e}}} \deg(\bar{v}_i)$. The next corollary and additional results in Appendix A-B follow immediately from this observation.

Corollary 2. *If all the variable-node degrees are even ($\Lambda_i \neq 0$ only for even i), $w_{\boldsymbol{s}}$ is even for every error vector $\boldsymbol{e}$.*

This "pathology" is important when considering a probabilistic analysis of $w_{\boldsymbol{s}}$ (as done in the rest of the paper), which due to approximations may deviate from this behavior. However, as we show in Appendix A-B, even a small fraction of odd-degree variable nodes is sufficient for mitigating this pathology.

B. Probabilistic Check-Node Perspective Analysis

The relation given in Eq. (1) becomes exponentially hard to evaluate as $w_{\boldsymbol{e}}$ grows, and therefore a probabilistic analysis is required from the check-node perspective.

Let $u_j \in \mathcal{U}$ be a check node corresponding to the syndrome element s_j. Let $\mathcal{V}(u_j)$ be the subset of variable nodes connected to u_j. We say that u_j is *unsatisfied* if $s_j = 1$, which occurs if an odd number of nodes $v_i \in \mathcal{V}(u_j)$ are erroneous.

Lemma 3. *(From [16]) the probability of a check node u_j of degree d being unsatisfied is*

$$p_u(d) = p_{\mathrm{o}}(d, p) = \frac{1}{2}\left(1 - (1-2p)^d\right). \tag{2}$$

Proof: For each $v_i \in \mathcal{V}(u_j)$, an error occurs with probability p, statistically independent of other nodes. By its definition, $p_{\mathrm{o}}(|\mathcal{V}(u_j)| = d, p)$ describes the probability of an odd number of such errors. ∎

The probability distribution of $w_{\boldsymbol{s}}$ can be derived from the probability of unsatisfied checks. Let $\mathcal{U}_{\mathrm{u}} = \{u_j \in \mathcal{U} : s_j = 1\}$ be the subset of unsatisfied check nodes. Then,

$$\mathrm{P}(w_{\boldsymbol{s}} = k) = \mathrm{P}\left(|\mathcal{U}_{\mathrm{u}}| = k\right). \tag{3}$$

In general, the simultaneous satisfaction of multiple check nodes is governed by statistically dependent events. This complicates the explicit derivation of $\mathrm{P}(w_{\boldsymbol{s}})$. The approximate independence assumption made in [16] is not accurate enough for our purposes (as shown numerically later in Section VII). Moreover, the approximated normal distribution with exact variance approach proposed in [17] is prohibitive for our purposes, since the two parameters describing the distribution, both depending on p, make the ML estimation untractable. We will therefore use a special class of codes for estimation, under which the events are *strictly independent*. We refer to such codes as *non-intersecting variable nodes* (NIV) codes. The rationale behind this terminology, together with an analysis of the conditions and properties of these codes, is presented in Section IV-B. Although this special class of codes exhibits poor error-correction performance, it provides an *exact* probability distribution of the syndrome weight, thus highly useful for estimation, as discussed in the next corollaries (we show in the next section how to add error correction to NIV codes via bi-layer concatenation).

Corollary 4. *For a check-regular code $\mathcal{C}$ with $\Omega(x) = x^{d_c}$ and NIV, $w_{\boldsymbol{s}}$ is a Binomial RV with m trials and probability $p_{\mathrm{u}}(d_c)$.*

Corollary 5. *For a code $\mathcal{C}$ with general $\Omega(x)$ and NIV, $w_{\boldsymbol{s}}$ is a Poisson-Binomial RV, with probability vector $\boldsymbol{p} \in (0,1)^m$ composed of $m\Omega_i$ values equal to $p_{\mathrm{u}}(i)$, for $1 \le i \le d_c$.*

Notice that for check-regular codes, all the entries of the vector $\boldsymbol{p}$ from Corollary 5 are $p_{\mathrm{u}}(d_c)$, and the Poisson-Binomial distribution degenerates to the Binomial distribution from Corollary 4.

IV. In-Memory Error Control Architecture

In this section we describe the proposed architecture for efficient and reliable in-memory computations. We formalize the use of an LDPC code for estimation, and detail the construction allowing to combine an estimation NIV code within a general error-correcting code.

A. Proposed Architecture and Operation

We employ and analyze an LDPC coding scheme for reliable in-memory computations. Let $\mathcal{C}_1, \mathcal{C}_2$ be two codes with PCM $\mathcal{H}_1, \mathcal{H}_2$, respectively, designed such that $\mathcal{C}_2 \subset \mathcal{C}_1$ (meaning that $\mathcal{H}_1$ is a sub-matrix containing $m_1 < m_2$ rows

from $\mathcal{H}_2$, and any codeword of $\mathcal{C}_2$ is also a codeword of $\mathcal{C}_1$). $\mathcal{C}_1$ and $\mathcal{C}_2$ will be referred to as the *estimation code* and *correction code*, respectively. These codes are operated in the proposed architecture as in the following stages:

1) **Encode:** let $\boldsymbol{x} \in \{0,1\}^{n-m_2}$ be an information word to be stored. Encode $\boldsymbol{x}$ using $\mathcal{C}_2$ to obtain $\boldsymbol{c}$, and store $\boldsymbol{c}$.
2) **Estimation:** perform a periodical *in-memory computation* of $\boldsymbol{s} = \mathcal{H}_1(\boldsymbol{c}')^T$, that is, the syndrome of the possibly corrupted $\boldsymbol{c}'$ with respect to $\mathcal{C}_1$. Estimate $\hat{p}$ (discussed in Section V), and whether p exceeds some predefined tolerance p_{tol} (discussed in Section VI). If it does, move to step 3.
3) **Decoding:** read $\boldsymbol{c}'$ from memory into a resourceful processing unit, use $\mathcal{C}_2$ to decode $\boldsymbol{c}'$ into $\boldsymbol{c}$, and store $\boldsymbol{c}$ again.

As can be seen, the estimation code $\mathcal{C}_1$ is used for in-memory BER estimation and detection, while the correction code $\mathcal{C}_2$ is used for out-of-memory error correction. The correction with $\mathcal{C}_2$ is also performed upon standard read accesses, when the data is required externally. The efficiency of the proposed scheme is based upon two principles:

1) Ensuring that the resource-expensive out-of-memory correction will occur as rarely as possible, only when it must, so that high IMC efficiency will be maintained.
2) Ensuring that once correction is invoked, the BER will be low enough so it will succeed with very high probability.

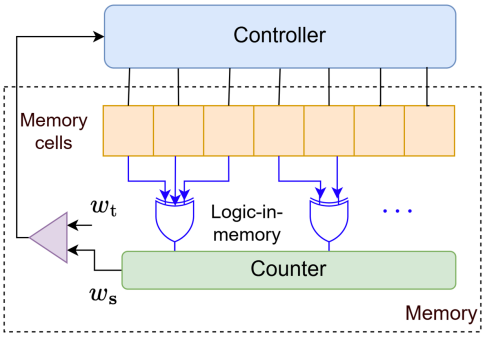

Fig. 1: System block diagram of the proposed architecture.

Fig. 1 illustrates the proposed architecture and operation. The XOR gates (blue) are determined through the check equations in $\mathcal{H}_1$. The comparator of the syndrome weight, calculated by a counter accumulating logical '1' XOR results, with the a predefined threshold w_{t} (discussed in Section VI) tells the controller whether a correction invocation is required. The controller then uses the full $\mathcal{H}_2$ to decode and re-writes the data to the memory cells. It can be seen that each memory cell is connected to exactly one XOR gate. This corresponds to degree-1 variable nodes in $\mathcal{H}_1$, which is very useful for both in-memory routing and the derivation of ML estimator. The latter is discussed in the following section.

B. NIV Codes: Independent Syndrome Elements

As discussed in Section III-B, we wish for the estimation code $\mathcal{C}_1$ that the satisfaction of different syndrome elements will form statistically independent events, whereas in general this is not the case.

Observation 6. *For a memoryless channel, the random events $(s_j = 1, s_k = 1)$ are statistically independent if and only if $\mathcal{V}(u_j) \cap \mathcal{V}(u_k) = \emptyset$.*

Therefore, we seek codes that eliminate the dependency through disjoint variable nodes.

Definition 7. *A code $\mathcal{C}$ is said to have **non-intersecting variable nodes (NIV)** if for every $l \in \{2, \ldots, m\}$,*

$$\forall\, 1 \le j_1 < \cdots < j_l \le m \;:\; \bigcap_{i=1}^{l} \mathcal{V}(u_{j_i}) = \emptyset. \tag{4}$$

Following Observation 6, we see that for NIV codes the satisfaction of syndrome elements constitutes independent events as desired.

NIV codes can be explicitly described through the variable DD, as shown in the next lemma.

Proposition 8. *A code $\mathcal{C}$ has NIV if and only if $\Lambda(x) = x$, that is, the degree of any variable node is 1.*

Proof: Let v be a variable node of degree $d > 1$. Then, for $\mathcal{U}(v) = u_1, \ldots, u_d$, we have $\bigcap_{i=1}^{d} \mathcal{V}(u_i) = \{v\}$. In the other direction, assume there exist u_1, u_2 such that $\mathcal{V}(u_1) \cap \mathcal{V}(u_2) \neq \emptyset$, then for $v \in \mathcal{V}(u_1) \cap \mathcal{V}(u_2)$ we have $\deg(v) \ge 2$. ■

This unusual restriction on the variable DD has substantial implication on the code performance for error correction.

Proposition 9. *A code $\mathcal{C} \in \mathbf{C}^n(x, \Omega)$, thus NIV, has $\frac{m}{2}\Omega''(1)$ codewords of weight 2.*

Proof: Beginning with the all-zero codeword, for each $u \in \mathcal{U}$, flipping any pair $v_1, v_2 \in \mathcal{V}(u)$ to 1 provides a codeword (since $\deg(v_i) = 1$). For any u of degree i there are $\binom{i}{2}$ such pairs, and there are $m\Omega_i$ such nodes. Therefore,

$$\#\text{ weight 2 words} = m \sum_{i=2}^{d_c} \Omega_i \frac{i(i-1)}{2} = \frac{m}{2}\Omega''(1).$$

■

We conclude that NIV codes have poor error correction capability, which is entirely prohibitive for using them in the traditional framework as error detection and correction codes. However, the structure of the proposed architecture from Section IV-A allows the use of NIV codes for estimation only, while also taking *sub-codes* of them having much more powerful correction capability. This is formalized in the following definition, in which we denote by $[A; B]$ the vertical concatenation of two matrices A and B with the same number of columns.

Definition 10. *an **estimation-correction code** is a pair $\mathcal{C}_1, \mathcal{C}_2$ of LDPC codes that hold*

1) *$\mathcal{C}_2 \subset \mathcal{C}_1$, that is, $\mathcal{H}_2 = [\mathcal{H}_1; \bar{\mathcal{H}}_2]$,*
2) *$\mathcal{C}_1 \in \mathbf{C}^n(x, \Omega)$ has NIV,*
3) *$\mathcal{C}_2 \in \mathbf{C}^n(\Lambda, \Omega)$, has desired variable DD and threshold.*

In addition to NIV codes enabling an accurate analysis of the probability distribution of $w_{\boldsymbol{s}}$, they introduce another significant advantage – since each variable node is of degree 1, each memory cell should only be electrically connected

to a single parity check logic, thus preventing a complex in-memory routing between cells and parity checks. A random code construction for an estimation-correction code from Definition 10 is now described. The bottom-up approach is to start from the estimation code $\mathcal{C}_1$ and add check equations to nest it in a correction code $\mathcal{C}_2$ with desired DD.

Construction 1. (bi-layer estimation-correction code)
1) Construct $\mathcal{C}_1$: given a check DD $\Omega(x)$, choose a random code $\mathcal{C}_1 \in \mathbf{C}^n(x,\Omega)$ (e.g. using Richardson-Urbanke socket-based construction [11]) to obtain $\mathcal{H}_1$.
2) Construct $\bar{\mathcal{C}}_2$: given a variable DD $\Lambda(x)$ with $\Lambda_1 = 0$, set $\bar{\Lambda}(x) = \Lambda(x)/x = \sum_{i=2}^{d_v} \Lambda_i x^{i-1}$. Choose a random code $\bar{\mathcal{C}}_2 \in \mathbf{C}^n(\bar{\Lambda},\Omega)$. Vertically concatenate the PCM $\bar{\mathcal{H}}_2$ of $\bar{\mathcal{C}}_2$ to $\mathcal{H}_1$ from the previous step to obtain $\mathcal{H}_2$.

Discussion. *Let $\bar{m}_2$ denote the number of rows in $\bar{\mathcal{H}}_2$. Note that $m_1 = n/\Omega'(1)$. Moreover, $\bar{\Lambda}'(1) = \sum_{i=2}^{d_v} \Lambda_i(i-1) = \Lambda'(1) - 1$, and therefore $\bar{m}_2 = n(\Lambda'(1)-1)/\Omega'(1)$. Hence, we have $m_2 = m_1 + \bar{m}_2 = n\Lambda'(1)/\Omega'(1)$. In addition, since $\mathcal{H}_1$ adds a single 1 to each column of $\bar{\mathcal{H}}_2$, the variable DD corresponding to $\mathcal{H}_2$ is $\bar{\Lambda}(x)\cdot x = \Lambda(x)$, as desired.*

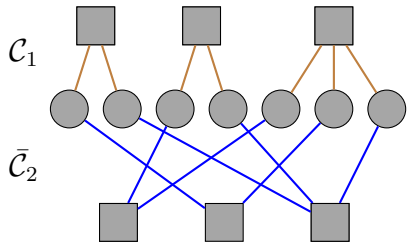

Fig. 2: Illustration of a bi-layer estimation-correction code, with DD $\Omega(x) = 2/3\cdot x^2 + 1/3\cdot x^3$ and $\Lambda(x) = x^2$, and codes $\mathcal{C}_1 \in \mathbf{C}^7(x,\Omega)$ (upper checks) and $\mathcal{C}_2 \in \mathbf{C}^7(\Lambda,\Omega)$ (all checks). Notice that in this example $\bar{\Lambda}(x) = x$, inducing $\bar{\mathcal{C}}_2 \in \mathbf{C}^7(x,\Omega)$ as well (lower checks).

Fig. 2 illustrates the Tanner graph of a bi-layer estimation-correction code from Construction 1 with $\Lambda(x) = x^2$, $\Omega(x) = \frac{2}{3}x^2 + \frac{1}{3}x^3$. Construction 1 is an instance of the known bi-layer LDPC codes [24], specifically composed of an estimation layer (checks of $\mathcal{C}_1$) and a complementary layer (checks of $\bar{\mathcal{C}}_2$), to be used along with the estimation layer, for powerful error correction. Efficient design techniques have been proposed for bi-layer codes in [25]. A topic for future work is studying such bi-layer designs, and constructing such $\mathbf{C}^n(\Lambda,\Omega)$ codes with desired correction performance, in particular, allowing $\mathcal{C}_1$ and $\bar{\mathcal{C}}_2$ to have different check-degree distributions.

V. Maximum Likelihood BER Estimation

The probabilistic relations in Section III-B suggest the use of statistical estimation of p from $w_{\boldsymbol{s}}$. Since no a-priori assumptions are made for p, the framework of maximum likelihood (ML) estimation is most suitable.

A. Estimation for Check-Regular Codes

For completeness, we revisit a result from [16], using our terminology and notations. Following Corollary 4, a ML estimator can be derived for check-regular codes.

Proposition 11. *(from [16]) For an NIV check-regular code $\mathcal{C} \in \mathbf{C}^n(x, x^{d_c})$, the BER ML estimator is*

$$\hat{p}_{\mathrm{reg}}(w_{\boldsymbol{s}}) = \frac{1}{2}\left(1 - ([1 - 2w_{\boldsymbol{s}}/m]_+)^{1/d_c}\right), \tag{5}$$

where $[x]_+ \triangleq \max\{0,x\}$.

Proof: See Appendix B-A. ∎

Note that we used $\Lambda(x) = x$ to require NIV and ensure that $w_{\boldsymbol{s}}$ is strictly Binomial. This is in contrast with [16], in which the Binomial distribution is an approximation for general $\Lambda(x)$.

B. Estimation for Check-Irregular Codes

We now wish to generalize the result from check-regular to any general irregular check DD. However, due to the complex combinatorial form of its probability mass function (PMF), the Poisson-Binomial distribution leads to an intractable ML estimation. We therefore begin by using a widely used Poisson approximation to the Poisson-Binomial distribution [26]. From this point forward we denote $\zeta_p \triangleq 1 - 2p$.

Definition 12. *For an NIV check-irregular code $\mathcal{C} \in \mathbf{C}^n(x,\Omega)$, define*

$$\lambda(p) \triangleq \boldsymbol{p}^T\mathbf{1} = \sum_{i=1}^{d_c} m\Omega_i p_{\mathrm{u}}(i) = \frac{m}{2}\left(1 - \Omega(\zeta_p)\right),$$

with $\boldsymbol{p}$ as defined in Corollary 5 and $\mathbf{1}$ being the all-one vector. Moreover, define

$$\tilde{p}_{\mathrm{u}}(\Omega) \triangleq \frac{\lambda(p)}{m} = \frac{1}{2}\left(1 - \Omega(\zeta_p)\right), \tag{6}$$

as the average unsatisfied-check probability.

Proposition 13. *Let $w_{\boldsymbol{s}}$ be a Poisson-Binomial RV as defined in Corollary 5, and W be a Poisson RV with parameter $\lambda(p)$. Let p satisfy $p_{\mathrm{u}}(d_c) \le \frac{1}{4}$ (Eq. (2)). Then,*

$$\begin{aligned}\|w_{\boldsymbol{s}} - W\|_1 &\le 8\left[\frac{1 - 2\Omega(\zeta_p) + \Omega(\zeta_p^2)}{1 - \Omega(\zeta_p)}\right] \\ &= 16p\left(1 + \Omega''(1)/\Omega'(1)\right) + \mathcal{O}\left(p^2\right),\end{aligned}$$

with $\|w_{\boldsymbol{s}} - W\|_1 = \sum_{r=0}^{\infty} |P(w_{\boldsymbol{s}} = r) - P(W = r)|$.

Proof: See Appendix C. ∎

Based on Proposition 13, for small values of p, it is very reasonable to consider $w_{\boldsymbol{s}}$ from Corollary 5 as a Poisson RV. Since a Poisson RV W follows the concentration bound $\mathrm{P}\left(|W - \lambda| > x\right) \le \exp\left\{-\frac{x^2}{2(\lambda + x)}\right\}$ [27], it can be verified that the tail probability (for values that cannot be attained in Poisson-Binomial) satisfies

$$\mathrm{P}\left(w_{\boldsymbol{s}} > m\right) \le \exp\left\{-\frac{m}{2}\left(1 - \tilde{p}_{\mathrm{u}}(\Omega)\right)^2\right\}, \tag{7}$$

which is practically 0 for reasonable values of m (e.g. $m > 100$) and small enough p (for which $1 - \tilde{p}_{\mathrm{u}}(\Omega)$ is close to 1). Moreover, since the Binomial distribution in the check-regular case is a special case of the Poisson-Binomial distribution, the approximation is further motivated in the general case.

Assumption 14. *From here on, unless stated otherwise, we assume* $w_{\boldsymbol{s}} \sim Pois\left(\lambda(p)\right)$.

We can now derive the ML estimation for irregular codes.

Proposition 15. *Under Assumption 14, for an NIV check-irregular code* $\mathcal{C} \in \mathbf{C}^n(x,\Omega)$*, the BER ML estimator is*

$$\hat{p}_{\rm irr}(w_{\boldsymbol{s}}) = \frac{1}{2}\left(1-\Omega^{-1}\left([1-2w_{\boldsymbol{s}}/m]_+\right)\right), \tag{8}$$

where $\Omega^{-1}(x)$ *is the inverse function of the polynomial* $\Omega(x)$.

Proof: See Appendix B-B. ∎

We note that finding the inverse $\Omega^{-1}(\cdot)$ as an algebraic expression is generally not possible, but the inversion can be performed numerically for values in $[0,1]$.

Observation 16. *For a check-regular code* $\mathcal{C} \in \mathbf{C}^n(x,x^{d_c})$ *we have* $\Omega^{-1}(x) = x^{1/d_c}$*. Substituting to Eq.* (8) *we get Eq.* (5)*, showing that the check-regular estimator is a special case of the check-irregular estimator.*

C. Degree-Distribution-Dependent Estimation Performance

In the following, $\hat{p}$ and $\hat{p}(w_{\boldsymbol{s}})$ will be used as compact replacements of $\hat{p}_{\rm irr}(w_{\boldsymbol{s}})$ from Eq. (8). We now analyze the estimation performance of the proposed estimator in terms of MSE, that is,

$$\textbf{mse}(p) \triangleq \mathbb{E}\left[(\hat{p}-p)^2\right] = \textbf{var}(p) + \textbf{bias}(p)^2,$$

where $\textbf{var}(p) \triangleq \mathbb{E}\left[(\hat{p}-\mathbb{E}[\hat{p}])^2\right]$ and $\textbf{bias}(p) \triangleq \mathbb{E}\left[\hat{p}\right]-p$ are the estimator's variance and bias, respectively. Our main goal is to study the effect of $\Omega(x)$ on the MSE. We define $\tau \triangleq \mathrm{P}\left(w_{\boldsymbol{s}} \le m/2\right)$, so that $1-\tau$ is the *probability of truncation* of the estimator (that is, the case where the value inside the $[\cdot]_+$ in Eq. (8) is negative). τ, which is a function of m and p, can be easily calculated from the Poisson distribution assumed for $w_{\boldsymbol{s}}$. Using a similar concentration bound as in Eq. (7), it can be verified that

$$1-\tau \le \exp\left\{-m\left(\frac{1}{2}-\tilde{p}_{\rm u}(\Omega)\right)^2\right\}, \tag{9}$$

which indicates that the truncation becomes exponentially negligible as m grows. We denote $g(x) \triangleq 1-2\frac{x}{m}$, $\mathbb{E}_c[X] \triangleq \mathbb{E}[X|w_{\boldsymbol{s}} \le \frac{m}{2}]$.

Lemma 17. *It holds that*

$$\tau\mathbb{E}_c\left[\Omega^{-1}(g(w_{\boldsymbol{s}}))\right] = \zeta_p + \mathcal{O}\left(\max\left\{m(1-\tau),p^2\right\}\right).$$

Proof: See Appendix D-A. ∎

Corollary 18. *It follows from Lemma 17 that*

$$\begin{aligned}\textbf{bias}(p) &= \frac{\zeta_p}{2} - \frac{\tau}{2}\mathbb{E}_c\left[\Omega^{-1}(g(w_{\boldsymbol{s}}))\right] \\ &= \mathcal{O}\left(\max\left\{m(1-\tau),p^2\right\}\right).\end{aligned}$$

Corollary 18 shows that the estimator is asymptotically unbiased for small crossover probabilities. We note that the regularity conditions under-which the ML estimator is known to be asymptotically unbiased [28] (Section 10.6.2) do not apply to $\hat{p}(w_{\boldsymbol{s}})$ due to the truncation $[1-2w_{\boldsymbol{s}}/m]_+$, and hence Corollary 18 is important for establishing this. This result is also crucial in the proof of the following theorem, which is the main result of this section.

Theorem 19. *It holds that*

$$\textbf{mse}(p) = \frac{1-\Omega(\zeta_p^2)}{4m\Omega'(\zeta_p)^2} + (1-\tau)\left(\frac{\zeta_p}{2}\right)^2 + \eta, \tag{10}$$

where

$$\eta = \mathcal{O}\left(\max\{m^2(1-\tau)^2, p^4, \frac{1}{m^2}, \frac{1-\tau}{m}, p^2(1-\tau)\}\right),$$

which is small compared to the leading terms for small p *and large* m.

Proof: See Appendix D-B. ∎

Theorem 19 shows that the MSE arises from two major distinct error mechanisms. The first term stems from the usual estimation variance which dominates in the non-truncated region of the estimator. The second term, resulting from the estimation bias, is significant in the truncated region obtained with probability $(1-\tau)$, where the error is given by the constant $\left(\frac{1}{2}-p\right) = \frac{\zeta_p}{2}$ with zero variance.

Using Theorem 19 we now wish to investigate the main trends of MSE dependency on the DD $\Omega(x)$. $1-\tau$ is clearly monotone increasing with $\lambda(p)$ and therefore with p, and thus the MSE is dominated by each of the terms in a different region of p values. We first wish to establish the region in which the estimator is typically not truncated, that is, where the MSE is dominated by the first (variance) term.

Definition 20. *For a given* $\Omega(x)$ *(and* m*) we define*

$$p_{\rm cut}(\Omega) \triangleq p \;\; \textit{for which} \;\; \frac{1-\Omega(\zeta_p^2)}{4m\Omega'(\zeta_p)^2} = (1-\tau)\left(\frac{\zeta_p}{2}\right)^2,$$

as the cutoff p*, that is, the value of* p *for which the two MSE terms are equal (recall* $\zeta_p = 1-2p$*).*

We will focus on the region $p < p_{\rm cut}(\Omega)$, which we will call the *non-truncated region*. Far enough from the boundary in that region, $1-\tau$ is very close to 0 and the MSE is approximately $(1-\Omega(\zeta_p^2))/(4m\Omega'(\zeta_p)^2)$. We therefore define our goal as described in the next problem.

Problem 1. *Given a value* $p_{\max} \in (0, 0.5)$ *bounding the interval for estimation, set* $\zeta_{\max} \triangleq 1-2p_{\max}$*, and find a DD* $\Omega(x)$ *that solves,*

$$\begin{aligned}&\textit{minimize} && J(\Omega) \triangleq \left(1-\Omega(\zeta_{\max}^2)\right)/\Omega'(\zeta_{\max})^2 \\ &\textit{subject to} && p_{\rm cut}(\Omega) > p_{\max}\end{aligned}$$

Since Problem 1 is still hard to solve, we further simplify $J(\Omega)$. Using Taylor's expansion about $\zeta_p = 1$ ($p=0$), we have

$$\begin{aligned}\Omega(\zeta_p^2) &= 1 - 4p\Omega'(1) + 4p^2\left(\Omega'(1) + 2\Omega''(1)\right) + \epsilon_1, \\ \Omega'(\zeta_p)^{-2} &= \Omega'(1)^{-2} + 4p\Omega'(1)^{-3}\Omega''(1) + \epsilon_2,\end{aligned}$$

with $\epsilon_1 = \mathcal{O}(p^3)$, $\epsilon_2 = \mathcal{O}(p^2)$. Using algebraic manipulations we get

$$J(\Omega) = \frac{p_{\max}}{m}\left[\frac{1-p_{\max}}{\Omega^{'}(1)} + \frac{2p_{\max}}{\Omega^{'}(1)^2}\Omega^{''}(1)\right] + \mathcal{O}(p_{\max}^3).$$

We denote by $\tilde{J}(\Omega)$ this approximation of $J(\Omega)$ after neglecting $\mathcal{O}(p_{\max}^3)$ terms. We can now observe two important dependencies:

1) For the family of check-regular codes, with $\Omega^{'}(1) = d_c$ and $\Omega^{''}(1) = d_c(d_c-1)$, we have
$$\tilde{J}(\Omega) = \frac{p_{\max}}{m}\left[\frac{1-3p_{\max}}{d_c} + 2p_{\max}\right],$$
which is clearly monotone decreasing with the degree d_c.
2) For the family of codes with a given average degree $\Omega^{'}(1) = a_{\mathrm{R}}$, $\tilde{J}(\Omega)$ is monotone increasing with $\Omega^{''}(1)$. We denote $\mathbf{var}(\Omega) \triangleq \sum_{i=1}^{d_c} i^2\Omega_i - {a_{\mathrm{R}}}^2$ as the variance of the check DD. We get that $\Omega^{''}(1) = \mathbf{var}(\Omega) + a_{\mathrm{R}}(a_{\mathrm{R}}-1)$, and so we can see that for integral a_{R} the check-regular code of degree a_{R}, with $\mathbf{var}(\Omega) = 0$, minimizes $\tilde{J}(\Omega)$.

We conclude that for small values of $p_{\max}$ and under the constraint $p_{\mathrm{cut}}(\Omega) > p_{\max}$, check-regular codes minimize the dominant terms of the MSE for every integral average check degree. Moreover, increasing the degree decreases the MSE. However, $p_{\mathrm{cut}}(\Omega)$ also decreases when the degree d_c is increased (as will be seen in Section VII), and so for a given $p_{\max}$, a certain maximal degree exists for which the constraint $p_{\mathrm{cut}}(\Omega) > p_{\max}$ is satisfied. Therefore, a possible solution methodology is to calculate $p_{\mathrm{cut}}(\Omega)$ for check-regular codes with increasing degrees, and take the maximal degree for which the constraint is still satisfied. The effectiveness of this section's findings will be demonstrated in Section VII.

For completeness, we propose a design procedure for an estimation-correction code based on this section's results.

Procedure 1. Given some tolerable error rate $p_{\max}$ (e.g., based on the predetermined error resilience of some neural network using the data [29]), perform the following stages:

1) For the family of check-regular codes $\Omega(x) = x^{d_c}$, find the maximal d_c^* for which $p_{\mathrm{cut}}(\Omega) > p_{\max}$.
2) Draw a random code $\mathcal{C}_1 \in \mathbf{C}^n(x, x^{d_c^*})$.
3) Use some known procedure for optimizing $\Lambda(x)$ given $\Omega(x) = x^{d_c^*}$ and some desired correction threshold (see [30, Chapter 4.10.1] for example).
4) Draw a random code $\bar{\mathcal{C}}_2 \in \mathbf{C}^n(\bar{\Lambda}, x^{d_c^*})$.
5) Encode $\boldsymbol{x} \in \{0,1\}^{n-k_2}$, with $k_2 = n\cdot\left(1 - \frac{\Lambda^{'}(1)}{d_c^*}\right)$, using $\mathcal{C}_2$ obtained from concatenating the PCMs of $\mathcal{C}_1$ and $\bar{\mathcal{C}}_2$ from above. Use the check equations of $\mathcal{C}_1$ for in-memory estimation, and the entirety of $\mathcal{C}_2$'s equations for error correction outside of memory.

VI. Error-Threshold Detection

Recall that the architecture described in Section IV-A is based on an indication whether the current BER satisfies $p > p_{\mathrm{tol}}$. This can be achieved by invoking the BER estimator $\hat{p}(w_{\boldsymbol{s}})$ and comparing it to p_{tol}. However, a more direct approach to detecting the threshold crossing is by a hypothesis-testing framework pursued in this section. We emphasize that since $\mathcal{C}_2$ can be designed as a standard powerful code, our focus on the hypothesis testing remains on $\mathcal{C}_1$, used for *deciding when* to decode.

A. Hypothesis Testing for Threshold-Crossing Detection

Considering the proposed architecture, we aim at distinguishing between the region $p > p_{\mathrm{tol}}$ in which a correction *must* be invoked, and the region $p \leq p_{\mathrm{tol}}$ where correction is *not required* and wastes computation resources. Under this set-up, it is reasonable to introduce a margin $[p_{\mathrm{tol}} - \delta, p_{\mathrm{tol}})$ in which it is equally "acceptable" to invoke correction or not (not required, but less wasteful than when p is very low). This is formalized in the following definition.

Definition 21. *For $\delta \in (0, p_{\mathrm{tol}})$, a δ-gapped hypothesis testing for correction invocation is the binary set of hypotheses*

$$\begin{aligned} H_0 &: p \leq p_{\mathrm{tol}} - \delta \ , \text{ correction should not be invoked,} \\ H_1 &: p > p_{\mathrm{tol}} \qquad , \text{ correction should be invoked.} \end{aligned}$$

The gap δ is crucial for non-degenerate analysis of the DD-dependent detection performance when there is no prior knowledge regarding p, as we will see in Section VI-B. It is noted that given some non-trivial (non-uniform) prior, a Bayesian hypothesis testing without a gap can be considered, a topic which we leave for future research. We define $\psi(x) \triangleq \frac{m}{2}\left(1 - \Omega(1-2x)\right)$, and recall from Definition 12 that $\psi(p) = \lambda(p)$ is the mean of $w_{\boldsymbol{s}}$. Since $\Omega(x)$ is monotone increasing, so does $\psi(x)$, and it holds that $p > x$ if and only if $\lambda(p) > \psi(x)$. Therefore, we can reformulate the problem to testing whether $\lambda(p) \leq \psi(p_{\mathrm{tol}} - \delta)$ or $\lambda(p) > \psi(p_{\mathrm{tol}})$. Under Assumption 14, $w_{\boldsymbol{s}}$ is a Poisson random variable, which meets the conditions of Karlin-Rubin theorem [31]. Therefore, the test that rejects H_0 if and only if $w_{\boldsymbol{s}} > w_{\mathrm{t}} \in \{0, \ldots, m/2\}$ (w_{t} is a decision threshold) is uniformly most powerful[1]. The type-I and type-II error probabilities [28] are now given by

$$\begin{aligned} \alpha(\delta) &\triangleq \mathrm{P}(\text{reject } H_0 | H_0) = \mathrm{P}\left(w_{\boldsymbol{s}} > w_{\mathrm{t}} | \lambda(p) \leq \psi(p_{\mathrm{tol}} - \delta)\right), \\ \beta &\triangleq \mathrm{P}(\text{accept } H_0 | H_1) = \mathrm{P}\left(w_{\boldsymbol{s}} \leq w_{\mathrm{t}} | \lambda(p) > \psi(p_{\mathrm{tol}})\right), \end{aligned}$$

respectively. We denote by $F_{\mathrm{pois}}(x;\lambda)$ the cumulative distribution function (CDF) of a Poisson RV with parameter λ, and

$$\begin{aligned} \alpha_\psi(\delta, w_{\mathrm{t}}) &\triangleq 1 - F_{\mathrm{pois}}(w_{\mathrm{t}}; \psi(p_{\mathrm{tol}} - \delta)), \\ \beta_\psi(w_{\mathrm{t}}) &\triangleq F_{\mathrm{pois}}(w_{\mathrm{t}}; \psi(p_{\mathrm{tol}})). \end{aligned}$$

Since $\frac{d}{d\lambda}F_{\mathrm{pois}}(x;\lambda) = -\frac{d}{dx}F_{\mathrm{pois}}(x-1;\lambda) \leq 0$, taking $\alpha(\delta), \beta$ with equality in the conditioning on $\lambda(p)$ implies

$$\alpha(\delta) \leq \alpha_\psi(\delta, w_{\mathrm{t}}) \ , \quad \beta \leq \beta_\psi(w_{\mathrm{t}}). \tag{11}$$

Notice that α_ψ, β_ψ depend on p_{tol} as well, but since it is a system parameter not to be set in design, we omit it for brevity.

[1]The reader is referred to [28] for details on uniformly most powerful tests (Definition 8.3.11), the Karlin-Rubin theorem (Theorem 8.3.17) and its conditions. It is also noted that although that theorem is stated for hypotheses with no gap ($\delta = 0$), the proof also holds for any $\delta > 0$.

B. Optimizing Detection Performance

We are now interested in designing the check DD (uniquely determining $\psi(x)$) to optimize the hypothesis-testing performance. Following conventional ROC analysis methods [32], we aim to maximize $1-\beta-\alpha(\delta)$ for any point on the curve. Since this is hard to do we will use the well-known technique of optimizing a surrogate function [33].

Definition 22. *For a given parameter set $(\delta, w_{\mathrm{t}}, p_{\mathrm{tol}})$ and function $\psi(x)$, let*

$$D_{\psi}\left(\delta, w_{\mathrm{t}}\right) \triangleq 1-\beta_{\psi}(w_{\mathrm{t}})-\alpha_{\psi}(\delta, w_{\mathrm{t}}). \tag{12}$$

From Eq. (11) we have that $D_{\psi}(\delta, w_{\mathrm{t}}) \leq 1-\beta-\alpha(\delta)$. We will therefore use $D_{\psi}(\delta, w_{\mathrm{t}})$ as a closed-form lower bound surrogate for the detection performance, to be maximized. However, the relationship between D_{ψ} and a given check DD and its corresponding $\psi(x)$ is not immediate. We further clarify this relationship through the following results. Define $\Delta \triangleq \psi(p_{\mathrm{tol}})-\psi(p_{\mathrm{tol}}-\delta)$, and

$$D^{*}(\Delta) \triangleq \sum_{j=0}^{w_{\mathrm{t}}} \frac{e^{-\psi(p_{\mathrm{tol}})}}{j!}\left[e^{\Delta}\left(\psi(p_{\mathrm{tol}})-\Delta\right)^{j}-\psi(p_{\mathrm{tol}})^{j}\right].$$

Theorem 23. *For every set $(\delta, w_{\mathrm{t}}, p_{\mathrm{tol}})$ such that $\delta \in (0, p_{\mathrm{tol}})$ and $w_{\mathrm{t}} \leq \psi(p_{\mathrm{tol}}-\delta)$, it holds that*

1) *$D^{*}(\Delta) = D_{\psi}(\delta, w_{\mathrm{t}}) \leq 1-\beta-\alpha(\delta)$,*
2) *$D^{*}(\Delta)$ is monotone increasing with Δ.*

Proof: By definition

$$\begin{aligned} D_{\psi}(\delta, w_{\mathrm{t}}) &= F_{\mathrm{pois}}(w_{\mathrm{t}}; \psi(p_{\mathrm{tol}}-\delta)) - F_{\mathrm{pois}}(w_{\mathrm{t}}; \psi(p_{\mathrm{tol}})) \\ &= \sum_{j=0}^{w_{\mathrm{t}}}\left[e^{-\psi(p_{\mathrm{tol}}-\delta)}\frac{\psi(p_{\mathrm{tol}}-\delta)^{j}}{j!} - e^{-\psi(p_{\mathrm{tol}})}\frac{\psi(p_{\mathrm{tol}})^{j}}{j!}\right] \\ &= \sum_{j=0}^{w_{\mathrm{t}}}\frac{e^{-\psi(p_{\mathrm{tol}})}}{j!}\left[e^{\Delta}\left(\psi(p_{\mathrm{tol}})-\Delta\right)^{j}-\psi(p_{\mathrm{tol}})^{j}\right], \end{aligned}$$

which is exactly $D^{*}(\Delta)$. Moreover,

$$\frac{dD^{*}}{d\Delta} = \sum_{j=0}^{w_{\mathrm{t}}}\frac{e^{-\psi(p_{\mathrm{tol}})}}{j!}\left[e^{\Delta}\left(\psi(p_{\mathrm{tol}})-\Delta\right)^{j}\left(\psi(p_{\mathrm{tol}})-\Delta-j\right)\right],$$

in which all terms are strictly positive for $j < w_{\mathrm{t}} \leq \psi(p_{\mathrm{tol}}-\delta) = \psi(p_{\mathrm{tol}})-\Delta$, and non-negative for $j = w_{\mathrm{t}}$. ∎

Remark 24. *The condition $w_{\mathrm{t}} \leq \psi(p_{\mathrm{tol}}-\delta)$ can be thought in terms of setting a threshold p_{t} on $\hat{p}(w_{\boldsymbol{s}})$ that satisfies $w_{\mathrm{t}} = \psi(p_{\mathrm{t}})$, and then requiring $p_{\mathrm{t}} \leq p_{\mathrm{tol}}-\delta$, i.e., setting the threshold below the gap region, which intuitively maximizes the utility of the gap region in reducing β without contributing to α.*

Lemma 25. *For every $0 < \delta < p_{\mathrm{tol}}$ it holds that*

$$\Delta = m\delta\left[\Omega^{'}(1)-(2p_{\mathrm{tol}}-\delta)\Omega^{''}(1)\right]+\mathcal{O}\left(mp_{\mathrm{tol}}^{3}\right). \tag{13}$$

Proof: See Appendix E. ∎

Theorem 23 along with Lemma 25 show that the surrogate lower bound on the detection performance can be maximized by maximizing the single parameter Δ, which in turn is described explicitly in terms of the check DD $\Omega(x)$. More specifically, by fixing δ, p_{tol} and neglecting the $\mathcal{O}(mp_{\mathrm{tol}}^{3})$ term, we have the following:

1) For the family of check-regular codes, the check degree that formally maximizes Δ is
$$d_{c}^{*} = \arg\max_{d}\left\{d-(2p_{\mathrm{tol}}-\delta)d(d-1)\right\} = \frac{1+2p_{\mathrm{tol}}-\delta}{4p_{\mathrm{tol}}-2\delta}. \tag{14}$$
Note that d_{c}^{*} is not necessarily integral, hence the term "formal" used above.
2) For the family of codes with a given average degree $\Omega^{'}(1) = a_{\mathrm{R}}$, the code that maximizes Δ is the code with minimal $\Omega^{''}(1) = \mathbf{var}(\Omega) + a_{\mathrm{R}}(a_{\mathrm{R}}-1)$. For integral a_{R} this code is the check-regular code with check degree a_{R}.

These put together suggest that check-regular codes with degree d_{c}^{*} maximize the surrogate lower bound when d_{c}^{*} is integral.

C. Setting the Detection Threshold

After setting the ROC curve by designing $\psi(x)$, it is left to choose a working point on that curve by setting w_{t}. Since β represents the critical event of not invoking correction when the BER exceeds its tolerance, it is reasonable to set some β^{*} as the maximum allowed β.

Definition 26. *Given $\psi(x)$ and the parameters $(\delta, p_{\mathrm{tol}})$, let*

$$w_{\mathrm{t}}^{*} = \max\{w_{\mathrm{t}}\} \;\; s.t \;\; \beta_{\psi}(w_{\mathrm{t}}) \leq \beta^{*}.$$

Lemma 27. *The threshold w_{t}^{*} satisfies*

$$w_{\mathrm{t}}^{*} = \min_{w_{\mathrm{t}}}\{\alpha_{\psi}(\delta, w_{\mathrm{t}})\} \;\; s.t \;\; \beta_{\psi}(w_{\mathrm{t}}) \leq \beta^{*}, \tag{15}$$

and it holds that $\beta < \beta^{*}$.

Proof: $F_{\mathrm{pois}}(w_{\mathrm{t}}; \cdot)$ is clearly monotone increasing with w_{t}. Following Eq. (11), minimizing $\alpha_{\psi}(\delta, w_{\mathrm{t}})$ translates to maximizing w_{t}, and the constraint ensures $\beta < \beta^{*}$. ∎

Discussion. *Eq. (15) can be described by choosing the leftmost point (smallest α_{ψ}) on the ROC curve for which the true-positive probability $1-\beta_{\psi}$ is above $1-\beta^{*}$, which is the best working point on that curve under the constraint β^{*}.*

D. Code Design Procedure

We end this section by proposing a design procedure for estimation-correction with good error-threshold detection.

Procedure 2. Apply Procedure 1 with the following adjustments:

1) Replace $p_{\max}$ in Procedure 1 with p_{tol} from Definition 21.
2) Replace d_{c}^{*} from Procedure 1 with d_{c}^{*} from Eq. (14).
3) Set w_{t}^{*} based on Eq. (15).
4) Instead of estimating $\hat{p}$, compare $w_{\boldsymbol{s}}$ to w_{t}^{*} and decide whether to perform out-of-memory decoding.

VII. Numerical Simulations

In this section we perform numerical simulations to demonstrate the concepts derived in previous sections.

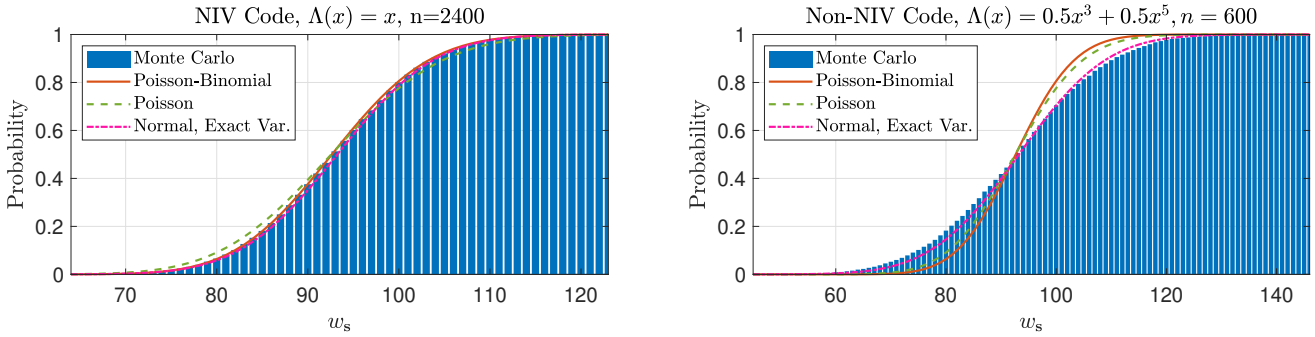

Fig. 3: Cumulative probability of $w_{\boldsymbol{s}}$, comparing Monte Carlo simulations (blue bars), Poisson-Binomial (solid red), Poisson (dashed green), and normal with corrected exact variance [17] (dash-dot magenta), for $\Omega(x) = x^6$, $p = 0.05$, $m = 400$, **left:** NIV code **right:** non-NIV code exhibiting worse fit by Poisson-Binomial model and Poisson approximation.

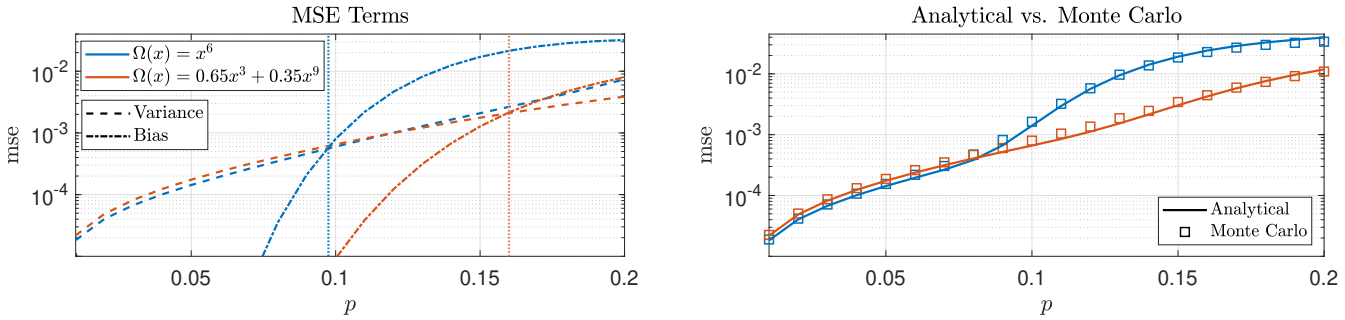

Fig. 4: **Left:** MSE terms (first and second terms from Theorem 19 - dashes and dash-dots, respectively), and $p_{\text{cut}}(\Omega)$ (vertical dotted lines), **right:** Analytical expression (solid) vs Monte Carlo simulations (squares), for different $\Omega(x)$ (colors) and $m = 100$.

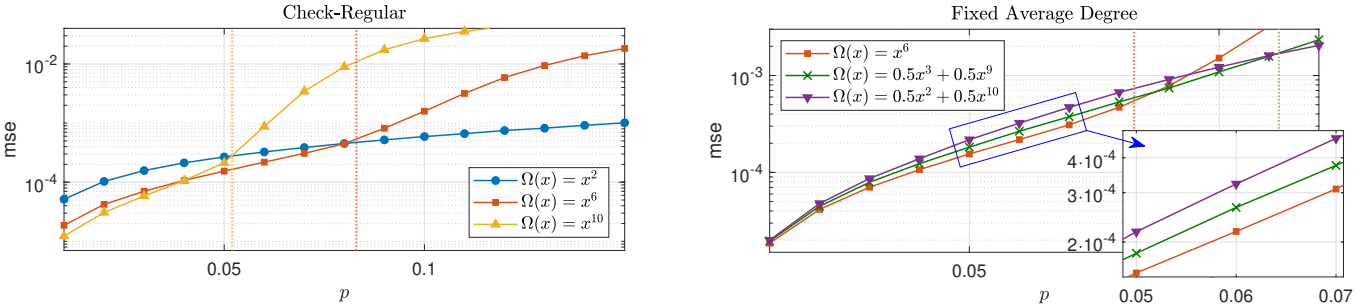

Fig. 5: **Left:** MSE for check-regular codes with different degrees, and $p_{\text{cut}}(\Omega)$ (vertical dotted lines), **Right:** MSE for fixed average degree with different distributions, $m = 100$.

A. Probabilistic Syndrome Weight

We first examine the probability distribution of $w_{\boldsymbol{s}}$. We randomly construct parity-check matrices using Richardson-Urbanke ensemble construction [11], and for each matrix we randomize error patterns of a BSC channel over the all-zero codeword, calculate the syndrome and its weight. We use this Monte Carlo approach to estimate the CDF of $w_{\boldsymbol{s}}$, and compare it to the Poisson-Binomial distribution (Corollary 5), the approximated Poisson distribution (Assumption 14), and the normal approximation with corrected exact variance proposed in [17]. Fig. 3 shows the results of this comparison for a NIV code and non-NIV code with $\Omega(x) = x^6$, $p = 0.05$ and $m = 400$. For both cases we have $\mathbb{E}[w_{\boldsymbol{s}}] \simeq 93.7$. It is first seen that the normal approximation with corrected exact variance from [17] captures the actual (Monte Carlo) distribution for both NIV and non-NIV cases, but since it leads to an impractical ML estimation we need the Poisson approximation. Considering that, it can be seen that the Poisson-Binomial distribution tightly fits the actual distribution (and the corresponding normal approximation) for NIV codes, whereas for non-NIV codes the actual distribution deviates with a larger variance, illustrating the importance of NIV for estimation. It is also seen that the Poisson distribution closely fits the accurate Poisson-Binomial one.

B. MSE

We now turn to examine the MSE of estimating p.

On the left side of Fig. 4, we show the first two MSE terms from Theorem 19 as a function of p, for two check DDs: $\Omega(x) = x^6$ and $\Omega(x) = 0.65x^3 + 0.35x^9$. The first term (dashes) originates from the non-truncated estimator variance, and the second term (dash-dotted) originates from the truncated bias. On the right side of Fig. 4, we compare the sum of the two terms (solid lines) to Monte Carlo simulations of the MSE, performed by drawing $w_{\boldsymbol{s}}$ from the Poisson-Binomial distribution for each value of p, evaluating $(\hat{p}(w_{\boldsymbol{s}}) - p)^2$ directly and taking the average. It is seen that the sum of these terms provides a very good estimate of the actual (Monte Carlo) MSE, and moreover, that the first term dominates for values of p smaller than $p_{\text{cut}}(\Omega)$ (Definition 20, shown as a vertical dotted line on the left-side figure), while the second term dominates for larger p.

In Fig. 5, we examine the MSE dependence on the DD $\Omega(x)$, as discussed in Problem 1 and thereafter. On the left side we show how increasing the degree of check-regular codes decreases the MSE for small enough values of p, but also decreases $p_{\text{cut}}(\Omega)$ thus limiting the "effective" region for estimation. On the right side, we show how for a given average check degree of $\Omega'(1) = 6$, the MSE increases when the DD variance $\mathbf{var}(\Omega)$ increases. These numerical results further validate the findings in Section V-C.

In Fig. 6 we demonstrate the importance of using NIV codes, by comparing the MSE from Eq. (10) to Monte Carlo simulations for NIV codes and non-NIV codes (general case in [16], here $\Lambda(x) = 0.5x^4 + 0.5x^6$), with the same $m = 100$ and the same check DDs $\Omega(x) = x^9$ and $\Omega = x^5$. It can be seen that unlike the NIV case (squares), in the non-NIV case (triangles) the actual MSE turns out to be significantly larger compared to the analytical expression.

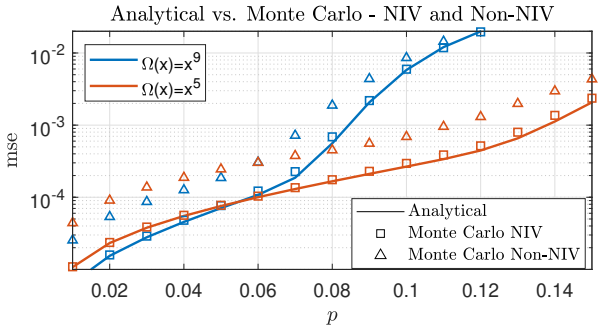

Fig. 6: MSE analytical expression (solid) vs Monte Carlo simulation for NIV codes (squares) and non-NIV codes (triangles).

Moreover, since the MSE highly depends on m, in Fig. 7 we verify that Eq. (10) provides a good estimate for different values of m, namely, $\{30, 100, 1000\}$. It can be well seen that the analytical expression fits the Monte Carlo results, and that the trends of variance-dominated and bias-dominated regions apply to all examined parameters.

C. Hypothesis Testing

We lastly examine the dependence of the hypothesis-testing performance on the DD $\Omega(x)$, as discussed following Theorem 23 and Lemma 25. In Fig. 8, we show the ROC curve of

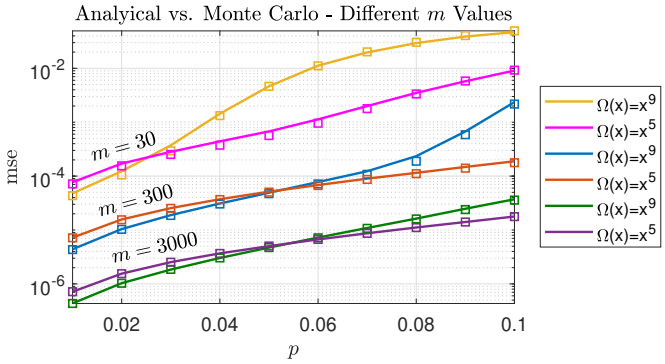

Fig. 7: MSE analytical expression (solid) vs Monte Carlo simulations (squares) for different m values and different DDs.

$1 - \beta_\psi(w_{\text{t}})$ as a function of $\alpha_\psi(\delta, w_{\text{t}})$, for $p_{\text{tol}} = 10^{-2}, \delta = 10^{-3}$ and $m = 100$, under which the degree d_c^* that formally maximizes Δ in (14) is 25.5.

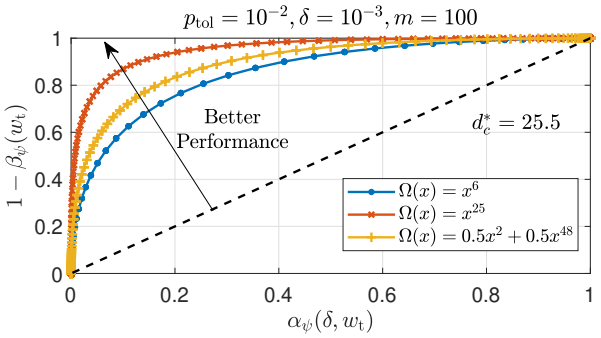

Fig. 8: ROC curve of the hypothesis testing for correction invocation, for different DDs.

Recall that the performance measure is $1 - \beta_\psi - \alpha_\psi$, which translates to the vertical distance from the $1 - \beta_\psi = \alpha_\psi$ line (dashed). It can therefore be seen that the performance is improved when increasing the check-regular degree from 6 (dot markers) to 25 (x markers), and also that the latter is superior to an irregular code with the same average degree of 25 (+ markers), due to the negative effect of the DD variance $\mathbf{var}(\Omega)$ analyzed above. These numerical results further validate the findings in Section VI-B.

VIII. Conclusions

In this paper we look into the problem of efficiently estimating the BER of stored data directly in-memory. This methodology enables an efficient error control, where the memory performs simple logic-in-memory to decide whether (costly) error correction is actually required. We employed a bi-layer LDPC code design, where only the first layer of degree-1 variable nodes is used for in-memory estimation, and a second layer, complementing the code to a desired variable degree distribution. We study a maximum-likelihood estimation of the BER from the syndrome weight of the first layer, and analyze the degree-distribution-dependent estimation performance. We then introduce a gapped hypothesis testing framework for

setting the syndrome-weight threshold. We analyze the degree-distribution-dependent hypothesis-testing performance. In both problems, check-regular codes are shown (for low crossover probabilities) to offer performance advantages.

IX. Acknowledgments

The authors wish to thank the JSAIT reviewers for valuable comments, and Prof. Ido Tal for valuable suggestions.

Appendix A
Variable-Node Perspective Analysis of $w_{\boldsymbol{s}}$

A. Proof of Theorem 1

The following will be useful in the proof ahead.

Definition 28. *Let $S^N = \{S_1, S_2, ..., S_N\}$ be a multiset of N sets S_i. The N-ary* symmetric difference *of S is defined by*

$$\Delta S^N \triangleq \left\{ \bigcup_{([N]^l)} \left(\bigcup_{j=1}^{l} S_{i_j} \right) : l \text{ odd} \right\},$$

that is, the multiset of all subsets of odd number of sets of S.

We note that Definition 28 is a straight-forward generalization of the known symmetric difference between two sets [34], $\Delta S^2 = S_1 \Delta S_2 \triangleq (S_1 \setminus S_2) \cup (S_2 \setminus S_1)$, through $\Delta S^N = (((S_1 \Delta S_2)\Delta S_3)\Delta \ldots \Delta S_N)$.

Lemma 29. *The N-ary symmetric difference's cardinality is*

$$\left|\Delta S^N\right| = \sum_{l=1}^{N} (-2)^{l-1} \sum_{([N]^l)} \left| \bigcap_{j=1}^{l} S_{i_j} \right|. \tag{16}$$

Proof: We prove through mathematical induction. Base: for $N = 2$, we have the known relation [34] $\left|\Delta S^2\right| = |S_1| + |S_2| - 2|S_1 \cap S_2|$. Step: we assume that Eq. (16) holds for $N-1$, and examine its right-hand term for N.

$$\begin{aligned}
&\sum_{l=1}^{N} (-2)^{l-1} \sum_{([N]^l)} \left| \bigcap_{j=1}^{l} S_{i_j} \right| \\
&\overset{(a)}{=} \sum_{l=1}^{N} (-2)^{l-1} \sum_{([N-1]^{l-1})} \left| \bigcap_{j=1}^{l-1} S_{i_j} \cap S_N \right| \\
&\quad + \sum_{l=1}^{N-1} (-2)^{l-1} \sum_{([N-1]^{l})} \left| \bigcap_{j=1}^{l} S_{i_j} \right| \\
&\overset{(b)}{=} |S_N| - 2 \sum_{l=2}^{N} (-2)^{l-2} \sum_{([N-1]^{l-1})} \left| \bigcap_{j=1}^{l-1} S_{i_j} \cap S_N \right| + \left|\Delta S^{N-1}\right| \\
&\overset{(c)}{=} |S_N| - 2\left|\Delta S^{N-1} \cap S_N\right| + \left|\Delta S^{N-1}\right| \overset{(d)}{=} \left|\Delta S^N\right|,
\end{aligned}$$

where (a) is obtained by splitting the intersections $\left|\bigcap_{j=1}^{l} S_{i_j}\right|$ to those that include S_N (first term) and those that do not, (b) is by excluding $l = 1$ from the first sum (resulting in $|S_N|$) and by the induction assumption on the second term, (c) is by replacing $l' \leftarrow l - 1$ and observing that the intersection with S_N in each subset is the same as the intersection of the entire set with S_N, and (d) is from the known relation of Eq. (16) for $N = 2$. ∎

The proof of Theorem 1 now directly follows.

Proof: (Theorem 1) A syndrome element satisfies $s_j = 1$ if and only if the corresponding check u_j is connected to an odd number of nodes in $\mathcal{V}_{\boldsymbol{e}}$. We therefore need to evaluate the cardinality of the symmetric difference of the multiset $\{\mathcal{U}(\bar{v}_1), \ldots, \mathcal{U}(\bar{v}_{w_{\boldsymbol{e}}})\}$. The expression for $w_{\boldsymbol{s}}$ now follows directly from Lemma 29, completing the proof. ∎

B. Asymptotic Even-Odd Symmetry of $w_{\boldsymbol{s}}$

Following Theorem 1, we examine the asymptotic even-odd properties of $w_{\boldsymbol{s}}$.

Corollary 30. *If all the variable-node degrees are odd ($\Lambda_i \neq 0$ only for odd i), the parities of $w_{\boldsymbol{s}}$ and $w_{\boldsymbol{e}}$ are the same.*

This behavior is important when considering codes with all-even variable-node degrees, for which we will always see an even syndrome weight. However, the following results show that an asymptotic even-odd symmetry is obtained for every other type of codes.

Proposition 31. *For a code with all-odd variable-node degrees and an error weight $w_{\boldsymbol{e}}$ that is Binomially distributed with a fixed parameter $p \in (0, 0.5)$, we have*

$$\lim_{n\to\infty} \Pr\{w_{\boldsymbol{s}} \text{ is even}\} = \lim_{n\to\infty} \Pr\{w_{\boldsymbol{s}} \text{ is odd}\} = \frac{1}{2}.$$

Proof: Following Corollary 30, $w_{\boldsymbol{e}}$ and $w_{\boldsymbol{s}}$ have the same parity. Since $w_{\boldsymbol{e}}$ is a Binomial RV, we have

$$\lim_{n\to\infty} \Pr\{w_{\boldsymbol{s}} \text{ is odd}\} = \lim_{n\to\infty} p_{\mathrm{o}}(n, p) = \frac{1}{2},$$

since $(1-2p)^n \overset{n\to\infty}{\longrightarrow} 0$ for $p \in (0, 0.5)$. ∎

Proposition 32. *For every fraction $\alpha \in (0, 1)$ of variable nodes having odd degrees, and any fixed weight $w_{\boldsymbol{e}}$, it holds that*

$$\begin{aligned}
\lim_{n\to\infty} \Pr\{w_{\boldsymbol{s}} \text{ is even}\} &= \frac{1}{2}\left(1 + (1-2\alpha)^{w_{\boldsymbol{e}}}\right), \\
\lim_{n\to\infty} \Pr\{w_{\boldsymbol{s}} \text{ is odd}\} &= \frac{1}{2}\left(1 - (1-2\alpha)^{w_{\boldsymbol{e}}}\right).
\end{aligned}$$

Proof: The number $k \in \{0, \ldots, w_{\boldsymbol{e}}\}$ of odd-degree variable nodes in $\mathcal{V}_{\boldsymbol{e}}$ has hypergeometric distribution with population size n, sample size $w_{\boldsymbol{e}}$, and αn success states. It is well known that the hypergeometric probability mass function (PMF) with a fixed success fraction α converges to the Binomial PMF with probability α as $n \to \infty$ (see, e.g. [35]). Based on Theorem 1, $w_{\boldsymbol{s}}$ is odd if and only if k is odd, and therefore, $\lim_{n\to\infty} \Pr\{w_{\boldsymbol{s}} \text{ is odd}\} = p_{\mathrm{o}}(w_{\boldsymbol{e}}, \alpha)$. ∎

It can be seen from Lemma 32 that as $w_{\boldsymbol{e}}$ grows, the probabilities of $w_{\boldsymbol{s}}$ being even and odd both tend to $1/2$, implying an asymptotic symmetry of the parity of $w_{\boldsymbol{s}}$. Moreover, the limiting results in Lemma 32 also hold approximately when $w_{\boldsymbol{e}} = \mathcal{O}(pn)$ with a small fraction p (and not a constant as above), since the hypergeometric PMF is well approximated by the Binomial PMF in that case too [36]. Therefore, when $w_{\boldsymbol{e}}$ is Binomially distributed with a fixed small parameter $0 \leq p \ll 0.5$, we again obtain a full symmetry between odd and even weight as $n \to \infty$.

APPENDIX B
MAXIMUM LIKELIHOOD DERIVATIONS

A. Proof of Proposition 11 - Check-Regular ML

Proof: $p_u(d_c;p)$ is bijective and monotone increasing in $p \in [0,0.5]$. Hence, $p_u(d_c;\hat{p}) = p_u^* = \arg\max_{p_u} \mathrm{P}(w_{\boldsymbol{s}}|p_u)$ and $\hat{p}$ can be calculated from p_u^*. Since $w_{\boldsymbol{s}}$ is Binomial, we have

$$\frac{\partial}{\partial p_u}\log \mathrm{P}(w_{\boldsymbol{s}}=j|p_u) = \frac{j}{p_u} - \frac{m-j}{1-p_u} = \frac{j-mp_u}{p_u(1-p_u)},$$

and by setting to zero we get $j = mp_u$. Replacing j with $w_{\boldsymbol{s}}$ and requiring $p_u \leq 1/2$, we get $p_u(d_c;\hat{p}) = \frac{1}{2}\left(1-(1-2\hat{p})^{d_c}\right) = w_{\boldsymbol{s}}/m$, when $w_{\boldsymbol{s}} \leq m/2$. Taking the inverse provides the desired expression. For $w_{\boldsymbol{s}} > m/2$, we have $\frac{\partial}{\partial p_u}\log \mathrm{P}(w_{\boldsymbol{s}}|p_u) > 0$ (because $p \leq 1/2 \Rightarrow p_u \leq 1/2$), in that case the boundary value $p = 1/2$ maximizes the likelihood. ∎

B. Proof of Proposition 15 - Check-Irregular ML

Proof: Since $\Omega(x)$ is a polynomial with non-negative coefficients, it is clearly monotone increasing in $[0,1]$. Therefore, $\lambda(p)$ is also bijective and monotone increasing in $p \in [0,0.5]$, and $\lambda(\hat{p}) = \lambda^* = \arg\max_{\lambda} \mathrm{P}(w_{\boldsymbol{s}}|\lambda)$. Based on Assumption 14, we have

$$\frac{\partial}{\partial \lambda}\mathrm{P}(w_{\boldsymbol{s}}=j|\lambda) = e^{-\lambda}\frac{\lambda^{j-1}}{j!}\left(j-\lambda\right).$$

Replacing j with $w_{\boldsymbol{s}}$ and requiring $\lambda \leq m/2$ (since $p \leq 0.5$), we get $\lambda(\hat{p}) = \frac{m}{2}\left(1-\Omega(1-2\hat{p})\right) = w_{\boldsymbol{s}}$ when $w_{\boldsymbol{s}} \leq m/2$. Taking the inverse provides the desired expression. For $w_{\boldsymbol{s}} > m/2$, we have $\frac{\partial}{\partial \lambda}\mathrm{P}(w_{\boldsymbol{s}}|\lambda) > 0$, and again in that case the boundary value $p = 0.5$ maximizes the probability. ∎

APPENDIX C
PROOF OF PROPOSITION 13 - PROBABILITY-DISTRIBUTIONS DISTANCE

Proof: Following [26], if $\max_j\{p_j\} \leq \frac{1}{4}$ (where p_j are entries of $\boldsymbol{p}$), then $\|w_{\boldsymbol{s}} - W\|_1 \leq 16\sum_j p_j^2 / \sum_j p_j$. Since $p_{\mathrm{u}}(d)$ is monotone increasing with d, it suffices to require $p_{\mathrm{u}}(d_c) \leq \frac{1}{4}$. We know that the denominator is $\lambda(p) = \frac{m}{2}(1-\Omega(1-2p))$. The numerator is given by $\sum_{i=1}^{d_c} m\Omega_i p_{\mathrm{u}}^2(i) = \frac{m}{4}\sum_{i=1}^{d_c}\Omega_i\left[1-2(1-2p)^i + ((1-2p)^2)^i\right]$ and we get the desired expression. To show the limiting result, we use the Taylor expansions

$$\begin{aligned}\Omega(1-2p) &= \sum_{\ell=0}^{\infty}\frac{\Omega^{(\ell)}(1)}{\ell!}(-2p)^{\ell},\\ \Omega((1-2p)^2) &= \sum_{\ell=0}^{\infty}\frac{\Omega^{(\ell)}(1)}{\ell!}((1-2p)^2-1)^{\ell},\end{aligned}$$

with $\Omega^{(\ell)}(x)$ denoting the order-ℓ derivative of $\Omega(x)$, and get

$$\begin{aligned}\|w_{\boldsymbol{s}} - W\|_1 &\leq 8\cdot\frac{4[\Omega^{'}(1)+\Omega^{''}(1)]p^2 + \sum_{\ell=3}^{\infty}a_\ell p^\ell}{2\Omega^{'}(1)p + \sum_{\ell=2}^{\infty}b_\ell p^\ell}\\ &= 8p\cdot\frac{4[\Omega^{'}(1)+\Omega^{''}(1)] + \sum_{\ell=3}^{\infty}a_\ell p^{\ell-2}}{2\Omega^{'}(1) + \sum_{\ell=2}^{\infty}b_\ell p^{\ell-1}},\end{aligned}$$

where we denoted $\{a_\ell\},\{b_\ell\}$ as the finite coefficients of p^ℓ. Since the limits of the nominator and denominator of the second term exist (and different from zero), we have that $\|w_{\boldsymbol{s}} - W\|_1 = 16p\left(1+\Omega^{''}(1)/\Omega^{'}(1)\right) + \mathcal{O}\left(p^2\right)$. ∎

APPENDIX D
ESTIMATION PERFORMANCE

A. Proof of Lemma 17 - asymptotically unbiased estimator

Proof: We denote

$$\begin{aligned}\Omega^{-(i)}\left(g(k)\right) &\triangleq \frac{d^i}{dk^i}\Omega^{-1}\left(g(k)\right)\\ &= \left(\frac{-2}{m}\right)^i\frac{d^i}{dg(k)^i}\Omega^{-1}\left(g(k)\right).\end{aligned}$$

Using Taylor's expansion about $w_{\boldsymbol{s}} = 0$, restricting $w_{\boldsymbol{s}} \leq m/2$ we get the expectation

$$\tau\mathbb{E}_c\left[\Omega^{-1}\left(g(w_{\boldsymbol{s}})\right)\right] = \tau\sum_{i=0}^{\infty}\frac{\Omega^{-(i)}(1)}{i!}\mathbb{E}_c\left[{w_{\boldsymbol{s}}}^i\right]. \tag{17}$$

Since $\Omega^{'}(1) > 0$, by the inverse function theorem [37] $\Omega^{-1}(g(k))$ is analytic in the entire range $k \in [0, m/2]$, and therefore have a radius of convergence of $m/2$. By the Cauchy coefficient estimate [38], it holds that $\left|\frac{\Omega^{-(i)}(1)}{i!}\right| = \mathcal{O}\left(\left(\frac{2}{\Omega^{'}(1)\cdot m}\right)^i\right)$. Moreover, using Taylor's expansion of $\Omega(1-2p)$ about $p=0$, we have

$$\lambda(p) = \frac{m}{2}\left(1-\Omega(1-2p)\right) = m\Omega^{'}(1)\cdot p + \mathcal{O}(m\cdot p^2).$$

Now, since $\mathbb{E}_c[{w_{\boldsymbol{s}}}^i] < \mathbb{E}[{w_{\boldsymbol{s}}}^i] = \mathcal{O}(\lambda(p)^i)$ [39], the i-th term in Eq. (17) follows a geometric decay at least as fast as $\mathcal{O}\left((2p)^i\right)$, and so we have

$$\begin{aligned}\tau\mathbb{E}_c\left[\Omega^{-1}\left(g(w_{\boldsymbol{s}})\right)\right] &= \tau\left[\Omega^{-1}(1) + \Omega^{-(1)}(1)\mathbb{E}_c[w_{\boldsymbol{s}}] + \mathcal{O}(p^2)\right]\\ &= \tau\left[1 - \frac{2\mathbb{E}_c[w_{\boldsymbol{s}}]}{m\Omega^{'}\left(\Omega^{-1}(1)\right)} + \mathcal{O}(p^2)\right],\end{aligned}$$

where we used $\Omega^{-1}(1) = 1$ (since $\Omega(1) = 1$), and where the second equality is due to the inverse function theorem. It now holds that

$$\begin{aligned}\tau\mathbb{E}_c[w_{\boldsymbol{s}}] &= \tau\sum_{k=0}^{m/2}\mathrm{P}\left(w_{\boldsymbol{s}}=k|w_{\boldsymbol{s}}\leq\frac{m}{2}\right)\cdot k\\ &= \tau\sum_{k=0}^{m/2}\frac{\mathrm{P}(w_{\boldsymbol{s}}=k, w_{\boldsymbol{s}}\leq m/2)}{\mathrm{P}(w_{\boldsymbol{s}}\leq m/2)}\cdot k\\ &= \sum_{k=0}^{m}\mathrm{P}(w_{\boldsymbol{s}}=k)\cdot k - \sum_{k=m/2+1}^{m}\mathrm{P}(w_{\boldsymbol{s}}=k)\cdot k\\ &= \mathbb{E}[w_{\boldsymbol{s}}] - (1-\tau)\sum_{k=m/2+1}^{m}\frac{\mathrm{P}(w_{\boldsymbol{s}}=k, w_{\boldsymbol{s}}>m/2)}{1-\tau}\cdot k\\ &= \mathbb{E}[w_{\boldsymbol{s}}] - (1-\tau)\mathbb{E}[w_{\boldsymbol{s}}|w_{\boldsymbol{s}}>m/2]\\ &= \lambda(p) + \mathcal{O}(m(1-\tau))\\ &= m\Omega^{'}(1)\cdot p + \mathcal{O}\left(\max\left\{m(1-\tau), p^2\right\}\right).\end{aligned}$$

Plugging to the equation above we get $\tau\mathbb{E}_c\left[\Omega^{-1}\left(g(w_{\boldsymbol{s}})\right)\right] = 1-2p+\mathcal{O}\left(\max\left\{m(1-\tau),p^2\right\}\right)$. ∎

B. Proof of Theorem 19 - MSE closed form

The following lemma will be useful for the proof.

Lemma 33. *Given some smooth function $f : \mathbb{R} \to \mathbb{R}$ and a random variable X, it holds that*

$$\textbf{\textit{var}}(f(X)) = [f'(\mathbb{E}[X])]^2\textbf{\textit{var}}(X) - \frac{[f''(\mathbb{E}[X])]^2}{4}\textbf{\textit{var}}(X)^2 + \mathcal{O}\left(\left[f'''(\mathbb{E}[X])^2 + f'(\mathbb{E}[X])f''(\mathbb{E}[X])\right]\mathbb{E}[(X-\mathbb{E}[X])^3]\right)$$

Proof: We use Taylor's expansion for $f(X)$ about $\mathbb{E}[X]$ to get

$$f(X) = f(\mathbb{E}[X]) + f'(\mathbb{E}[X])(X-\mathbb{E}[X]) + \frac{f''(\mathbb{E}[X]}{2}(X-\mathbb{E}[X])^2 + \mathcal{O}\left(f'''(\mathbb{E}[X])(X-\mathbb{E}[X])^3\right),$$

and again for $f^2(X)$ to get

$$f^2(X) = f^2(\mathbb{E}[X]) + 2f(\mathbb{E}[X])f'(\mathbb{E}[X])(X-\mathbb{E}[X]) + \left[f'(\mathbb{E}[X])^2 + f(\mathbb{E}[X])f''(\mathbb{E}[X])\right](X-\mathbb{E}[X])^2 + \mathcal{O}\left(\left[f'''(\mathbb{E}[X])^2 + f'(\mathbb{E}[X])f''(\mathbb{E}[X])\right](X-\mathbb{E}[X])^3\right).$$

Taking the expectation of the latter and subtracting the expectation of the former squared completes the proof. ∎

We now move on to the proof of the theorem.

Proof: (Theorem 19) We denote $G(x) \triangleq \Omega^{-1}(g(x))$ and $\textbf{var}_c(X) \triangleq \textbf{var}(X|w_{\boldsymbol{s}} \leq \frac{m}{2})$. Following the expression in Lemma 17, we have

$$\textbf{bias}(p)^2 = \left(\frac{\zeta_p}{2}\right)^2 - \frac{\zeta_p}{2}\tau\mathbb{E}_c\left[G(w_{\boldsymbol{s}})\right] + \frac{\tau^2}{4}\mathbb{E}_c\left[G(w_{\boldsymbol{s}})\right]^2$$

∎

Next, for the estimation variance we observe

$$\begin{aligned}\mathbb{E}[\hat{p}^2] &= \sum_{k=0}^{m/2}\mathrm{P}(k)\left[\frac{1}{2} - \frac{1}{2}G(k)\right]^2 + \frac{1-\tau}{4} \\ &= \frac{1}{4} - \frac{1}{2}\sum_{k=0}^{m/2}\mathrm{P}(k)G(k) + \frac{1}{4}\sum_{k=0}^{m/2}\mathrm{P}(k)G(k)^2 \\ \mathbb{E}[\hat{p}]^2 &= \left[\frac{1}{2} - \frac{1}{2}\sum_{k=0}^{m/2}G(k)\right]^2 \\ &= \frac{1}{4} - \frac{1}{2}\sum_{k=0}^{m/2}\mathrm{P}(k)G(k) + \frac{1}{4}\left(\sum_{k=0}^{m/2}\mathrm{P}(k)G(k)\right)^2,\end{aligned}$$

and so we have

$$\begin{aligned}\textbf{var}(\hat{p}) &= \mathbb{E}[\hat{p}^2] - \mathbb{E}[\hat{p}]^2 \\ &= \frac{1}{4}\sum_{k=0}^{m/2}\mathrm{P}(k)G(k)^2 - \frac{1}{4}\left(\sum_{k=0}^{m/2}\mathrm{P}(k)G(k)\right)^2 \\ &= \frac{\tau}{4}\mathbb{E}_c\left[G(w_{\boldsymbol{s}})^2\right] - \frac{\tau^2}{4}\mathbb{E}_c\left[G(w_{\boldsymbol{s}})\right]^2 \\ &= \frac{\tau}{4}\textbf{var}_c\left(G(w_{\boldsymbol{s}})\right) + \frac{\tau-\tau^2}{4}\mathbb{E}_c\left[G(w_{\boldsymbol{s}})\right]^2.\end{aligned}$$

Combining the terms, the *MSE* follows

$$\begin{aligned}&\textbf{mse}(p) = \textbf{bias}(\hat{p}^2) + \textbf{var}(p) \\ &= \frac{\tau}{4}\left(\textbf{var}_c\left(G(w_{\boldsymbol{s}})\right) + \mathbb{E}_c\left[G(w_{\boldsymbol{s}})\right]^2\right) + \frac{\zeta_p^2}{4} - \frac{\zeta_p}{2}\tau\mathbb{E}_c\left[G(w_{\boldsymbol{s}})\right].\end{aligned}$$

Now, following Lemma 17 we write $\tau\mathbb{E}_c[G(w_{\boldsymbol{s}})] = \zeta_p + \eta_1$, where $\eta_1 = \mathcal{O}\left(\max\left\{m(1-\tau), p^2\right\}\right)$. We plug this into the equation above to get

$$\begin{aligned}&\textbf{mse}(p) \\ &= \frac{\tau}{4}\textbf{var}_c(G(w_{\boldsymbol{s}})) + \frac{\zeta_p^2}{4}\left(\frac{1}{\tau} - 1\right) + \zeta_p\frac{\eta_1}{2}\left(\frac{1}{\tau} - 1\right) + \eta_1^2.\end{aligned}$$

We notice that $\frac{1}{\tau} = 1 + (1-\tau) + \mathcal{O}((1-\tau)^2)$, and so

$$\textbf{mse}(p) = \frac{\tau}{4}\textbf{var}_c(G(w_{\boldsymbol{s}})) + (1-\tau)\frac{\zeta_p^2}{4} + \eta_2,$$

where $\eta_2 = \mathcal{O}\left(\max\{m^2(1-\tau)^2, p^4\}\right)$. Based on Lemma 33, we have

$$\textbf{var}_c\left(G(w_{\boldsymbol{s}})\right) = [\Omega^{-(1)}(g(\lambda))]^2\textbf{var}_c(w_{\boldsymbol{s}}) - \frac{[\Omega^{-(2)}(g(\lambda))]^2}{4}\textbf{var}_c(w_{\boldsymbol{s}})^2 + \mathcal{O}\left(m^{-3}\right),$$

where we used the known property of the third central moment of the Poisson distribution, $(w_{\boldsymbol{s}} - \mathbb{E}[w_{\boldsymbol{s}}])^3 = \lambda(p) = \mathcal{O}(m)$, and the fact that the power of m increases as twice the term order for following terms. As in the proof of Lemma 17, we notice that

$$\begin{aligned}\tau\mathbb{E}_c\left[{w_{\boldsymbol{s}}}^i\right] &= \mathbb{E}\left[{w_{\boldsymbol{s}}}^i\right] - (1-\tau)\mathbb{E}\left[{w_{\boldsymbol{s}}}^i | w_{\boldsymbol{s}} > \frac{m}{2}\right] \\ &= \mathbb{E}\left[{w_{\boldsymbol{s}}}^i\right](1 - \delta_{\mathbb{E}}(i)),\end{aligned}$$

with $\delta_{\mathbb{E}}(i) \triangleq (1-\tau)\mathbb{E}\left[{w_{\boldsymbol{s}}}^i | w_{\boldsymbol{s}} > \frac{m}{2}\right] / \mathbb{E}\left[{w_{\boldsymbol{s}}}^i\right] = \mathcal{O}(1-\tau)$. It now follows that

$$\begin{aligned}&\tau\textbf{var}_c(w_{\boldsymbol{s}}) \\ &= \tau\mathbb{E}_c[{w_{\boldsymbol{s}}}^2] - \tau\mathbb{E}_c[w_{\boldsymbol{s}}]^2 \\ &= \mathbb{E}[{w_{\boldsymbol{s}}}^2](1 - \delta_{\mathbb{E}}(2)) - \frac{1}{\tau}\left(\mathbb{E}[w_{\boldsymbol{s}}](1-\delta_{\mathbb{E}}(1))\right)^2 \\ &= \textbf{var}(w_{\boldsymbol{s}}) + \mathbb{E}[w_{\boldsymbol{s}}]^2\left(2\delta_{\mathbb{E}}(1) - (1-\tau)\right) - \delta_{\mathbb{E}}(2)\mathbb{E}[{w_{\boldsymbol{s}}}^2] \\ &= \textbf{var}(w_{\boldsymbol{s}}) + \Delta_{\mathbb{E}} + \mathcal{O}\left(m(1-\tau)\right),\end{aligned}$$

where $\Delta_{\mathbb{E}} \triangleq \left(2\delta_{\mathbb{E}}(1) - \delta_{\mathbb{E}}(2) - (1-\tau)\right)\lambda(p)^2 = \mathcal{O}\left((1-\tau)m^2p^2\right)$. Using the inverse function theorem, we get $\Omega^{-(1)}(g(\lambda)) = \frac{-2}{m\Omega'(\zeta_p)}$, and since $\textbf{var}(w_{\boldsymbol{s}}) = \mathcal{O}(m)$, we get

$$\tau\textbf{var}_c(G(w_{\boldsymbol{s}})) = \frac{4\textbf{var}(w_{\boldsymbol{s}})}{m^2\Omega'(\zeta_p)^2} + \eta_3,$$

where $\eta_3 = \mathcal{O}\left(\max\{m^{-2}, m^{-1}(1-\tau), p^2(1-\tau)\}\right)$. To increase the approximation accuracy, we use the actual Poisson-Binomial variance, to get

$$\begin{aligned}\textbf{var}(w_{\boldsymbol{s}}) &= \sum_{i=1}^{d_c} m\Omega_i p_u(i)(1-p_u(i)) = \frac{m}{4}\sum_{i=1}^{d_c}\Omega_i\left[1 - (\zeta_p^2)^i\right] \\ &= \frac{m}{4}\left[1 - \Omega(\zeta_p^2)\right],\end{aligned}$$

and so in total we get

$$\textbf{mse}(p) = \frac{1 - \Omega(\zeta_p^2)}{4m\Omega'(\zeta_p)^2} + (1-\tau)\left(\frac{\zeta_p}{2}\right)^2 + \eta,$$

where η takes the maximum between $\{\eta_i\}_{i=1}^3$.

APPENDIX E
PROOF OF LEMMA 25 - Δ EXPANSION

Proof: Expanding $\psi(x)$ to a Taylor series around $x = 0$ gives

$$\psi(x) = m\Omega^{'}(1)x - m\Omega^{''}(1)x^2 + \mathcal{O}(mx^3).$$

Therefore,

$$\begin{aligned}
&\psi(p_{\mathrm{tol}}) - \psi(p_{\mathrm{tol}} - \delta) \\
&\quad = m\Omega^{'}(1)\left[p_{\mathrm{tol}} - (p_{\mathrm{tol}} - \delta)\right] + m\Omega^{''}(1)\left[p_{\mathrm{tol}}^2 - (p_{\mathrm{tol}} - \delta)^2\right] \\
&\qquad + \mathcal{O}\left(m\left[p_{\mathrm{tol}}^3 - (p_{\mathrm{tol}} - \delta)^3\right]\right) \\
&\quad = m\Omega^{'}(1)\delta + m\Omega^{''}(1)\left(\delta^2 - 2p_{\mathrm{tol}}\delta\right) \\
&\qquad + \mathcal{O}\left(m \cdot \max_{0\le i\le 2, 0\le j\le 3-i}\left\{p_{\mathrm{tol}}^i \cdot \delta^j\right\}\right),
\end{aligned}$$

and since $\delta < p_{\mathrm{tol}}$, the last term can be expressed as $\mathcal{O}\left(mp_{\mathrm{tol}}^3\right)$. ∎

REFERENCES

[1] A. Sebastian, M. Le-Gallo, R. Khaddam-Aljameh, and E. Eleftheriou, "Memory devices and applications for in-memory computing," *Nature Nanotechnology, Vol. 15*, 2020.

[2] S. A. McKee, "Reflections on the memory wall," *Conference on Computing Frontiers*, 2004.

[3] S. Aga, S. Jeloka, A. Subramaniyan, S. Narayanasamy, D. Blaauw, and R. Das, "Compute caches," *IEEE International Symposium on High Performance Computer Architecture*, 2017.

[4] P. Deaville, B. Zhang, and N. Verma, "A fully row/column-parallel mram in-memory computing macro with memory-resistance boosting and weighted multi-column adc readout," *IEEE Journal of Solid-State Circuits, Vol. 60, No. 5*, 2025.

[5] S. Li, D. Niu, K. T. Malladi, H. Zheng, B. Brennan, and Y. Xie, "DRISA: A DRAM-based reconfigurable in-situ accelerator," *IEEE/ACM International Symposium on Microarchitecture*, 2017.

[6] J. Borghetti, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, and R. S. Williams, "'memristive' switches enable 'stateful' logic operations via material implication," *Nature, Vol. 464*, 2010.

[7] H. Mahmoudi, T. Windbacher, V. Sverdlov, and S. Selberherr, "Implication logic gates using spin-transfer-torque-operated magnetic tunnel junctions for intrinsic logic-in-memory," *Solid-State Electronics, Vol. 84*, 2013.

[8] C. L. Chen and M. Y. Hsiao, "Error-correcting codes for semiconductor memory applications: A state-of-the-art review," *IBM Journal of Research and development, Vol. 28*, 1984.

[9] O. Leitersdorf, B. Perach, R. Ronen, and S. Kvatinsky, "Efficient error-correcting-code mechanism for high-throughput memristive processing-in-memory," *IEEE Press*, 2021.

[10] R. Gallager, "Low-density parity-check codes," *IRE Trans. on Information Theory, Vol. 8, No. 1*, 1962.

[11] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. on Information Theory, Vol. 47, No. 2*, 2001.

[12] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. on Information Theory, Vol. 55, No. 7*, 2009.

[13] I. Tal and A. Vardy, "How to construct polar codes," *IEEE Trans. on Information Theory, Vol. 59, No. 10*, 2013.

[14] G. Zervakis, H. Saadat, H. Amrouch, A. Gerstlauer, S. Parameswaran, and J. Henkel, "Approximate computing for ML: State-of-the-art, challenges and visions," *Asia and South Pacific Design Automation Conference*, 2021.

[15] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," *IEEE European Test Symposium*, 2013.

[16] G. Lechner and C. Pacher, "Estimating channel parameters from the syndrome of a linear code," *IEEE Communications Letters, Vol. 17, No. 17*, 2013.

[17] C. Pacher, P. Grabenweger, and D. E. Simos, "Weight distribution of the syndrome of linear codes and connections to combinatorial designs," in *2016 IEEE International Symposium on Information Theory (ISIT)*, 2016.

[18] M. A. Shokrollahi, "New sequences of linear time erasure codes approaching the channel capacity," *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, 1999.

[19] O. Barak, D. Burshtein, and M. Feder, "Bounds on achievable rates of LDPC codes used over the binary erasure channel," *IEEE Trans. on Information Theory, Vol. 50, No. 10*, 2004.

[20] E. Paolini and M. Chiani, "On the threshold of right regular LDPC codes for the erasure channel," *IEEE 61st Vehicular Technology Conference*, 2005.

[21] P. Oswald and A. Shokrollahi, "Capacity-achieving sequences for the erasure channel," *IEEE Trans. on Information Theory, Vol. 48, No. 12*, 2002.

[22] R. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. on Information Theory, Vol. 27, No. 5*, 1981.

[23] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, fourth edition*. MIT Press, 2022.

[24] P. Razaghi and W. Yu, "Bilayer low-density parity-check codes for decode-and-forward in relay channels," *IEEE Trans. on Information Theory, Vol. 53, No. 10*, 2007.

[25] E. Ram and Y. Cassuto, "Design of bilayer and multi-layer LDPC ensembles from individual degree distributions," *IEEE Trans. on Information Theory, Vol. 67, No. 11*, 2021.

[26] L. H. Chen, "On the convergence of poisson binomial to poisson distributions," *The Annals of Probability*, 1974.

[27] C. Canonne. (2016) A short note on poisson tail bounds. [Online]. Available: https://www.cs.columbia.edu/~ccanonne/files/misc/2017-poissonconcentration.pdf

[28] G. Casella and R. L. Berger, *Statistical Inference, Second Edition*. Thomson Learning, 2002.

[29] R. M. Zur, Y. Jiang, L. L. Pesce, and K. Drukker, "Noise injection for training artificial neural networks: A comparison with weight decay and early stopping," *Medical physics*, vol. 36, no. 10, pp. 4810–4818, 2009.

[30] T. J. Richardson and R. L. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2007.

[31] S. Karlin and H. Rubin, "The theory of decision procedures for distributions with monotone likelihood ratio," *The Annals of Mathematical Statistics*, 1956.

[32] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters, Vol. 27, No. 8*, 2006.

[33] K. Lange, D. R. Hunter, and I. Yang, "Optimization transfer using surrogate objective functions," *Journal of computational and graphical statistics, Vol. 9, No. 1*, 2000.

[34] K. H. Rosen and K. Krithivasan, "Discrete mathematics and its applications," vol. 6, 1999.

[35] S. Ross, *A First Course in Probability*. Pearson Prentice Hall, 2010. [Online]. Available: https://books.google.co.il/books?id=Bc1FAQAAIAAJ

[36] U. Rupassara and B. Sedai, "On the convergence of hypergeometric to binomial distributions," *Computer and Information Science, Vol. 16, No. 3*, 2023.

[37] W. Rudin, *Real and complex analysis*. McGraw-Hill, Inc., 1987.

[38] E. M. Stein and R. Shakarchi, *Complex analysis*. Princeton University Press, 2010, vol. 2.

[39] J. Riordan, "Moment recurrence relations for binomial, poisson and hypergeometric frequency distributions," *Annals of Mathematical Statistics. 8 (2): 103–111.*, 1937.