

# A User-Independent Successive Interference Cancellation Based Coding Scheme for the Unsourced Random Access Gaussian Channel

Avinash Vem, Krishna R. Narayanan, Jean-Francois Chamberland and Jun Cheng

**Abstract**—This work introduces a novel coding paradigm for the unsourced multiple access channel model. The envisioned framework builds on a select few key components. First, the transmission period is partitioned into a sequence of sub-blocks, thereby yielding a slotted structure. Second, messages are split into two parts. A portion of the data is encoded using spreading sequences or codewords that are designed to be recovered by a compressed sensing type decoder. In addition to being an integral part of the data, the information bits associated with this first part also determine the parameters of the low-density parity check code employed during the subsequent stages of the communication process. The other portion of the message is encoded using the aforementioned low-density parity check code. The data embedded in this latter stage is decoded using a joint message passing algorithm designed for the  $T$ -user binary input real adder channel. Finally, devices repeat their codeword in multiple sub-blocks, with the transmission pattern being a deterministic function of message content independent of the identity of the device. When combined with successive interference cancellation, the ensuing communication infrastructure offers significant performance improvement compared to coding schemes recently published in the literature for unsourced random access.

**Index Terms**—Unsourced Multiple Access, Compressive Sensing, Successive Interference Cancellation,  $T$ -User Adder Channel, Forward Error Correction.

## I. INTRODUCTION

Recently, Polyanskiy introduced an original and timely multiple access problem termed the unsourced multiple access channel (MAC) model [1]. In this setting, a very large number of devices,  $K_{\text{tot}}$ , operate within a wireless network in an uncoordinated fashion. At any point in time, only a subset of these devices are active; we denote the cardinality of this subset by  $K_a$ . Every active device wishes to communicate a  $B$ -bit message to a central base station. Furthermore, the destination is solely interested in recovering the collection of sent messages without regard for the identity of individual

The work of K. R. Narayanan and J.-F. Chamberland is funded in part by the National Science Foundation under Grants CCF 1320924 and CCF 1619085. The work of Jun Cheng was supported in part by the Japan Society for the Promotion of Science through the Grant-in-Aid for Scientific Research (C) under Grant 18K04132, and in part by the Ministry of Internal Affairs and Communications Japan through the Strategic Information and Communications R&D Promotion Programme (SCOPE) under Grant 195007002.

K. R. Narayanan and J.-F. Chamberland are with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843, USA (emails: {krn, chmbrlnd}@tamu.edu). A. Vem is with Nvidia Corp., Santa Clara, CA, USA (email: vemavinash@tamu.edu). J. Cheng is with Doshisha University, Kyoto, Japan (email: jcheng@mail.doshisha.ac.jp).

A preliminary version of this paper was presented at the Information Theory Workshop, Kaohsiung, Taiwan, 2017.

sources. This scenario reflects emerging tasks such as distributed sensing and averaging. The payload  $B$  for unsourced scenarios is envisioned to be small, on the order of a hundred bits.

The unsourced, uncoordinated nature of the problem and its relation to large number of users represent a substantial departure from traditional multiple access channels. Consequently, this formulation engenders research questions in terms of fundamental limits and pragmatic designs with low-complexity coding schemes. For asymptotically long blocklengths, Chen, Chen and Guo studied this situation under the name of many-access channel and derived the capacity in [2]. However, our interest here is in short blocklengths. In this regime, Shannon rates and asymptotic information-theoretic limits only offer partial insight into the structure of the problem. Dispersion bounds and finite blocklength analysis are much more meaningful in this context, providing achievability benchmarks. In their initial treatment of the problem [1], Polyanskiy derives bounds on the performance of finite-length codes for this problem setting. These bounds serve as guidelines for practical communication schemes.

The development of efficient coding schemes tailored to the unsourced, uncoordinated MAC appears to be a challenging task. Most well-known, low-complexity coding solutions for the traditional MAC channel, such as code-division multiple access, rate-splitting [3], and interleave-division multiple access [4], implicitly assume some level of coordination and awareness across devices. For instance, parameters of an access scheme, like spreading sequences, code rates, time sharing schedules, and the Tanner graph of a code, are often device specific. That is, classic schemes explicitly or implicitly rely on the identity of the device being known to the destination. When message lengths are small, collecting and maintaining the information needed for such coordinated schemes become inefficient. This renders existing coding solutions for the traditional MAC inadequate for the unsourced MAC and, therefore, new paradigms are needed.

In their work [5], Ordentlich and Polyanskiy present the first low-complexity coding paradigm for the unsourced MAC. Within their proposed scheme, a transmission period is partitioned into smaller sub-blocks and devices randomly pick one sub-block to transmit their message. The encoding structure employed by each device is a concatenated code where the inner code is designed to recover the modulo- $p$  sum of codewords transmitted by devices, and the outer code aims at decoding multiple messages given the modulo- $p$  sum of their

codewords. Succinctly, the inner code operates in the spirit of integer-forcing [6], whereas the outer code is an optimal code for the  $T$ -user modulo- $p$  multiple access channel [7]. Under this construction, an underlying goal is to ensure that almost all sub-blocks contain at most  $T$  distinct signals.

While the contribution of Ordentlich and Polyanskiy is an important step in finding practical schemes for the unsourced MAC, there remains a substantial gap between the performance of their proposed scheme and the achievability limits derived in [1]. Indeed, they point to this gap and discuss avenues for improving performance. In [5, Section III.A], they discuss the possibility of improving their scheme by decoding the  $T$  messages using the real sum from the channel output, instead of first reducing the output of the channel to modulo- $p$  operations. Still, in an unsourced MAC, every device is forced to employ the same codebook. In view of this difficulty, they remark that “the task of designing low complexity capacity approaching same-codebook schemes for the real binary adder seems quite challenging.” Another significant limitation associated with their scheme, which is not mentioned in [5], stems from the fact that their approach is not amenable to interference cancellation. Thus, when more than  $T$  devices transmit in a same sub-block (slot), the collected information cannot be recovered and the sub-block content must be discarded altogether. As a result, their scheme must rely on a large number of sub-blocks to ensure that every device is received in a sub-block that contains at most  $T$  messages. This yields poor spectral efficiency.

This article proposes a novel pragmatic architecture for the unsourced, uncoordinated MAC channel. Our envisioned scheme uses compressive sensing, it leverages message bits to inform each device’s action with respect to channel access, and it enables successive interference cancellation. The ensuing paradigm substantially improves performance when compared to the state-of-the-art. We elaborate briefly on key features below, with an emphasis on components that allow us to circumvent some of the challenges present in [5]. (i) Active devices employ a same coding scheme; transmitted signals are determined solely by message bits and they are impervious to the identity of the device. More specifically, the message bits dictate parameters of the encoding process such as the interleaver and the spreading sequence. (ii) The proposed coding scheme is tailored to the binary-input, real-adder channel. The information message is split into two parts. The first portion governs the interleaver for an LDPC code, and the second part is encoded using this LDPC code. Bits associated with the first portion are communicated using a compressive sensing scheme. The second part is recovered using a message passing decoder that jointly recovers up to  $T$  messages within a sub-block. (iii) Active devices repeat their codewords in several sub-blocks, and the repetition patterns are set based on early message bits. This facilitates the application of successive interference cancellation across sub-blocks and, therefore, renders obsolete the over-provisioning of sub-blocks aimed at avoiding collisions with more than  $T$  messages per sub-block. With these contributions in mind, we proceed to formally describing our framework and its performance.

## II. SYSTEM MODEL

Let  $K_{\text{tot}}$  and  $K_a$  denote the total device count in the network and the number of active devices, respectively. In this system, each active device must send  $B$  bits of information within  $N$  uses of the channel. An alternate, insightful viewpoint that is especially relevant for the problem at hand is to order the  $M = 2^B$  possible messages, and then think of the decoding task as the base station having to identify the collection of message indices present within a round. Under this alternate representation, we define  $w_i \in [1 : M]$  as a random variable that denotes the message index of the  $i$ th active device. Throughout, we use  $[a : b]$  as a shorthand notation for the set of integers from  $a$  to  $b$ , including both endpoints. We assume that  $w_i$  is uniformly distributed over the set  $[1 : M]$ . Moreover, we postulate that the elements of  $\{w_i\}$  form a collection of independent random variables. We denote the set of transmitted message indices by  $\mathcal{W} = \{w_1, \dots, w_{K_a}\}$ .

Since the action of every active device is completely determined by the content of its message, we can represent the transmitted signal as  $\vec{X}(w)$ , where  $w \in [1 : M]$ . In our exposition, we assume that  $\vec{X}$  has dimension  $N$ , with symbol entries corresponding to the  $N$  uses of the channel. The observed signal vector at the destination can therefore be expressed as

$$\vec{Y} = \sum_{i \text{ is active}} \vec{X}(w_i) + \vec{Z}. \quad (1)$$

The transmission is subject to additive Gaussian noise, characterized by  $\vec{Z} \sim \mathcal{N}(0, \mathbf{I}_N)$ . For mathematical convenience, we introduce a notation for canonical boolean indicator functions  $\{s_i\}$ , where  $s_i = 1$  whenever device  $i$  is active and  $s_i = 0$  otherwise. We impose an average power constraint on the signal set,  $\frac{1}{M} \sum_{w=1}^M \|\vec{X}(w)\|^2 \leq NP$ . The decoder is tasked with producing a list of messages  $\mathcal{L}(\vec{Y}) = \{\hat{w}_1, \dots, \hat{w}_{K_a}\}$ . As in [5], system performance is evaluated in terms of the probability of error,

$$P_e = \max_{\substack{(s_1, \dots, s_{K_{\text{tot}}}) \in \{0,1\}^{K_{\text{tot}}} \\ \|(s_1, \dots, s_{K_{\text{tot}}})\|_0 = K_a}} \frac{1}{K_a} \sum_{i=1}^{K_{\text{tot}}} s_i \Pr(w_i \notin \mathcal{L}(\vec{Y})) \quad (2)$$

where  $\|\cdot\|_0$  denotes the Hamming weight. The objective is to design low-complexity encoding and decoding schemes such that  $P_e \leq \varepsilon$ , where  $\varepsilon$  is a target error probability.

Before advancing to the presentation of our algorithmic framework, it is instructive to draw a connection between the unsourced MAC problem and compressive sensing. Suppose that we gather all the possible signals as columns of a matrix, (1) can be transformed into

$$\vec{Y} = \underbrace{\begin{bmatrix} | & | & & | \\ \vec{X}(1) & \vec{X}(2) & \dots & \vec{X}(M) \\ | & | & & | \end{bmatrix}}_{\mathbf{X}} \underbrace{\begin{bmatrix} s_1 \\ \vdots \\ s_{K_{\text{tot}}} \end{bmatrix}}_{\vec{s}} + \vec{Z} \quad (3)$$

where  $\vec{s}$  is a  $K_a$ -sparse vector. That is, the unsourced MAC problem is an instance of noisy compressive sensing. Furthermore, one has the freedom to construct sensing matrix  $\mathbf{X}$ , as long as it fulfills the average power constraint. This may seem

encouraging given that much is known about compressive sensing [8], [9]. Unfortunately, a challenge with this viewpoint arises from the fact that the dimensionality of the solution space,  $2^B$ , far exceeds the capabilities of modern computing systems when  $B$  is approximately a hundred bits. Thus, we must turn to more pragmatic approaches.

Finally, a brief discussion of the unsourced nature of the problem is in order. While most random access protocols are de-facto unsourced, they use explicit or implicit feedback and/or handshaking during the contention phase to coordinate the resources assigned to each user. As a result, at the physical layer, codes can be designed assuming a certain deterministic interference profile. In this paper, we assume there is no feedback from the base station to allocate resources to the active users to manage interference. The lack of such coordination makes the design of the physical layer more challenging.

### III. PROPOSED SCHEME

The complexity-reduction framework we adopt in our proposed scheme combines recent advances in massive uncoordinated multiple access systems [10], [11], [12], compressive sensing [8], [9], and iterative decoding [13], [14], [15], [16]. The approach is based on a divide-and-conquer strategy, which seeks to reduce the number of interfering messages per sub-block and the dimensionality of the solution space for the compressive sensing sub-problems. Leveraging a connection between successive interference cancellation and message-passing decoding on bipartite graphs [10], [17], previous contributions have shown that random access can yield near optimal performance for uncoordinated systems [18], [19]. While traditional implementations often use shared randomness to generate schedules, a key insight of our proposed scheme is that the messages themselves, containing random bits, can be employed to create schedules. Given a smaller number of interfering messages per sub-block, the receiver can decode messages within lightly loaded sub-blocks, and employ successive interference cancellation to recover the sent information across sub-blocks. The overall architecture of the envisioned system appears in Fig. 1.

#### A. Encoding Process

The encoding process can be described in a hierarchical fashion. To begin, the  $N$  channel uses are divided into  $V$  sub-blocks (or sub-blocks), each of length  $N_s = N/V$ . The encoding operation at an active device takes place in two steps. Information  $w$  is first encoded into a sub-block codeword  $\vec{x}(w)$  of length  $N_s$ . This sub-block codeword is then transformed into signal  $\vec{X}(w)$  by repeating the coded sub-part over multiple sub-blocks as to create a bipartite graph amenable to successive interference cancellation; this architecture appears in Fig. 1. We elaborate on the separate steps below.

1) *Sub-Block Scheduler*: All devices share an identical codebook for sub-block encoding. Every encoder thus produces a sub-block codeword of length  $N_s$ . Mathematically, a device encodes message  $w$  into a codeword and subsequently creates modulated signal  $\vec{x}(w)$ . Since this process should be familiar to the reader, we often omit details and refer to  $\vec{x}(w)$

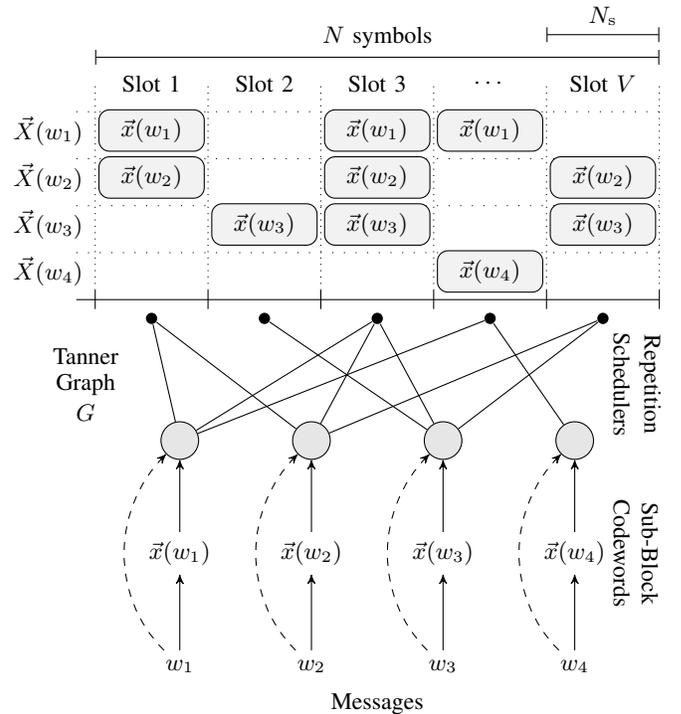


Fig. 1. This notional diagram depicts the operation of the system. Message  $w_i$  is (sub-block) encoded into codeword  $\vec{x}(w_i)$ , and then transmitted over several sub-blocks. The schedule across sub-blocks is a function of message content, and the information received at the destination within a sub-block is the aggregate sum of the codewords sent at that time.

as the transmitted codeword. This simplifies the discussion, and it should not lead to confusion as context can be employed to distinguish between a codeword and the corresponding modulated signal. Each device computes a repetition schedule  $g(w)$  using vector function  $g : [1 : M] \mapsto \{0, 1\}^V$ , where a transmission within sub-block  $t$  is indicated by a one at location  $t$ . Codeword  $\vec{x}(w)$  is then repeated  $\ell_w = \|g(w)\|_0$  times, with  $\ell_w$  sub-blocks being selected from  $[1 : V]$ . As illustrated in Fig. 1, a Tanner graph  $G$  can be employed to visualize the repetition pattern of the codewords. Variable nodes correspond to messages and check nodes are associated with sub-blocks. It is worth emphasizing that the transmission schedule and the number of repetitions,  $\ell_{w_i}$ , are deterministic functions of the message indices; they do not depend on the identity of the device. This is crucial because the receiver does not know the structure of the graph and, as such, it must be able to infer where to perform successive interference cancellation in real-time from successfully decoded sub-block codewords.

Once the connection between interference cancellation and message passing is revealed [10], the goal of the scheduler is to construct a graph that is conducive to iterative decoding with high probability [16]. To do so, we briefly review pertinent concepts. The degree of a left node is determined by  $\ell_w$ . Choosing  $w$  uniformly at random among message indices induces a distribution on  $\ell_w$  through function  $g(\cdot)$ . We denote the left degree distribution from the node perspective by  $L(\nu) = \sum_{k=1}^{\ell_{\max}} L_k \nu^k$ , where  $L_k$  denotes the fraction of the

device (left) nodes that are connected to  $k$  (right) sub-blocks. Similarly, we represent the left degree distribution from the edge perspective by  $\lambda(\nu) = \sum_{k=1}^{\ell_{\max}} \lambda_k \nu^{k-1}$ , where  $\lambda_k$  is the fraction of edges in  $G$  that are connected on the left to  $k-1$  other edges. The two distributions  $L(\nu)$  and  $\lambda(\nu)$  are linked by the equation  $\lambda(\nu) = \frac{L'(\nu)}{L'(1)}$ . The ensuing task is to choose mapping  $g(\cdot)$  such that a desired left degree distribution,  $L(\nu)$ , or equivalently  $\lambda(\nu)$ , is obtained. Luckily, much is known about generating good graphs [16]. We will revisit the notions introduced above and leverage established properties of random graphs when we build schedules for performance evaluation.

For a fixed sub-block scheduler, the collection of devices that transmit during sub-block  $j$  is determined by the messages they wish to communicate. We use  $\mathcal{N}_j$  to represent the collection of devices that transmit during sub-block  $j$ . Then, the signal received at the destination at that time becomes

$$\bar{y}_j = \sum_{i \in \mathcal{N}_j} \bar{x}(w_i) + \bar{z}_j, \quad (4)$$

where  $\bar{y}_j$  corresponds the segment of  $\bar{Y}$  send within sub-block  $j$  and, similarly,  $\bar{z}_j$  is the part of  $\bar{Z}$  matching sub-block  $j$ . Having established the slotted frame structure, we are ready to explore the specifics of sub-block encoding.

2) *Sub-Block Encoding*: The proposed transmission scheme for sub-blocks is composed of two parts: a sensing matrix for a  $T$ -sparse robust compressive sensing problem, and a channel code for the  $T$ -user binary-input real-adder channel. That is, the first few bits of every message are apportioned to a compressive sensing problem akin to (3), albeit with a smaller solution space. The residual bits from all the messages are recovered using an LDPC code. More specifically, the  $B$  information bits contained within a message are divided into two groups, one sequence of size  $B_p$ , which we call preamble bits, and another sequence of  $B_c = B - B_p$  bits referred to as coded bits. Let  $\bar{b}(w) \in \{0, 1\}^B$  be the binary representation of message  $w$ . Then, we can write  $\bar{b}(w) = (\bar{b}_p(w), \bar{b}_c(w))$ , where  $\bar{b}_p(w)$  and  $\bar{b}_c(w)$  denote the preamble and message bits within  $\bar{b}(w)$ , respectively. Likewise, let  $w_p$  be the index representation of  $\bar{b}_p(w)$ ; and  $w_c$ , that of the remaining bits  $\bar{b}_c(w)$ . For convenience, we introduce related quantities  $M_p = 2^{B_p}$  and  $M_c = 2^{B_c}$ . The essence of sub-block encoding is to transmit the  $B_p$  preamble bits via compressive sensing, and to employ a forward error correction code to transfer the trailing  $B_c$  message bits. For practical reasons which will soon become clear, we also use the preamble bits as side information to control the implementation of the linear code. Typically, one would want  $B_p \ll B_c$ .

The modulated sub-block signal  $\bar{x}(w)$  is formed as follows. For the preamble, let  $\mathbf{A} \in \{\pm\sqrt{P_p}\}^{N_p \times M_p}$  denote a compressive sensing matrix that can support the recovery of any  $T$  columns from an aggregate observation vector with high probability. The initial  $B_p$  symbols in  $\bar{x}(w)$  are obtained by taking the transpose of the  $w_p$ th column of  $\mathbf{A}$ , which we denote by  $\bar{a}_{w_p}^T$ . Constructing the second part of  $\bar{x}(w)$  is more involved. We begin with a good linear block code such as a low density parity check (LDPC) code or a spatially-coupled

low density parity check (SCLDPC) code of rate  $\frac{B_c}{N_s - N_p}$  and length  $N_c = N_s - N_p$ . As an example, we consider the case where codebook  $\mathcal{C}$  is chosen uniformly at random from an SCLDPC ensemble [20]. Let the modulated codewords associated with  $\mathcal{C}$  be denoted by  $\{\bar{c}(1), \dots, \bar{c}(M_c)\}$ , where  $\bar{c}(w_c) = (c_1(w_c), \dots, c_{N_c}(w_c)) \in \{\pm\sqrt{P_c}\}^{N_c}$ . Every codeword then satisfies power constraint

$$\|\bar{c}(w_c)\|_2^2 \leq N_c P_c. \quad (5)$$

We employ the preamble index  $w_p$  to permute the ordering of  $\bar{c}(w_c)$ . Let  $\tau: [1 : M_p] \mapsto [1 : N_c!]$  denote a hash function that maps these  $B_p$  preamble bits into  $[1 : N_c!]$ , an integer labeling of all possible  $N_c$ -permutations. Function  $\tau(w_p)$  can be used to choose a permutation  $\pi_{\tau(w_p)} \in S_{N_c}$ , which is applied to codeword  $\bar{c}(w_c) \in \mathcal{C}$  before it is modulated. Above,  $S_{N_c}$  designates the symmetric group [21]. Thus, the trailing part of  $\bar{x}(w)$  can be expressed as  $\pi_{\tau(w_p)}(\bar{c}(w_c))$ . Appending the modulated version of the permuted codeword to the coded preamble yields

$$\bar{x}(w) = \left( \bar{a}_{w_p}^T, \pi_{\tau(w_p)}(\bar{c}(w_c)) \right). \quad (6)$$

The codeword creation process is admittedly somewhat intricate and, as such, we summarize key features in Fig. 2.

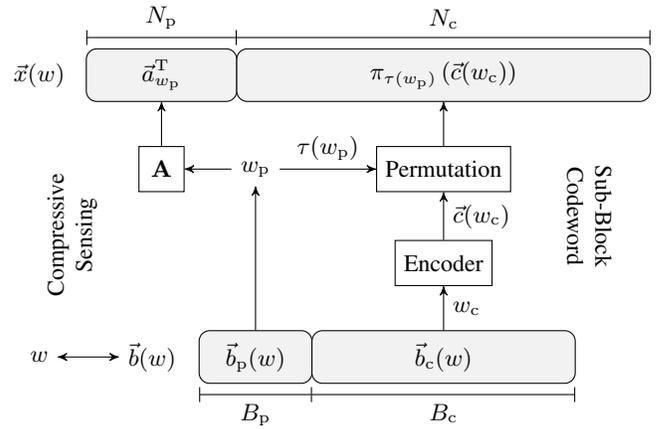


Fig. 2. For message  $w$  and corresponding bit sequence  $\bar{b}(w) = (\bar{b}_p(w), \bar{b}_c(w))$ , a device creates sub-block signal  $\bar{x}(w)$  using the schematic displayed above.

The purpose of permuting codewords is to decorrelate the multiple access interference from other devices even though they use an identical linear code. The reordering results in a performance that is similar to that observed when adopting different codes across devices, yet it simplifies encoding/decoding hardware significantly. This is similar to interleave-division multiple access scheme originally proposed in [4]. We note that the overall code is no longer linear because  $\bar{a}_{w_p}^T$  is prepended to the beginning of  $\bar{x}(w)$  and  $\bar{c}(w)$  is reordered. Still, once  $\bar{a}_{w_p}^T$  is recovered (along with  $w_p$ ) and removed from  $\bar{x}(w)$ , we can reverse the effects of the permutation on  $\bar{c}(w_c)$ . As we will see shortly, decoding can then be accomplished using a belief propagation decoder that works on the joint graph of all the received signals.

The use of compressed sensing in conjunction with a coding scheme also appears in the work of Chen, Chen and Guo in [2]

where compressed sensing is used to identify users and then a random codebook is used to convey the information. However, unlike in [2], our approach is completely unsourced and the user is not identified, rather the compressed sensing part is used to identify interleavers. As a result, in [2], the compressed sensing part is considered as a capacity penalty. In our case, it is not a capacity penalty - rather information is conveyed using the compressed sensing part too. Secondly, in [2], the compressed sensing part is used only once whereas in our case, each sub-block requires a compressed sensing part. Finally, in [2], information-theoretic bounds are derived assuming random coding whereas our work focuses on practical coding schemes.

The proposed scheme also bears some similarity to the scheme in [22] where a  $T$ -user multiple access code is used in conjunction with successive interference cancellation for user identification. The schemes in [22], [23], [24] use an outer code and physical layer network coding to decode modulo-sums of the transmitted codewords in order to implement the  $T$ -user multiple access code. However, in our paper, we do not decode the modulo sum, rather, the  $T$ -user multiple access code is implemented directly using LDPC codes.

Schemes that combine random access and compressive sensing for 5G wireless have been proposed in [25] and nice survey is presented in [26]. A scheme that uses compressive sensing for multiple access in sensor networks is presented in [27]. Our proposed scheme is substantially different from these schemes. Unlike [25], [26], [27], our approach uses compressed sensing only to convey the parameters of the LDPC code.

### B. Decoding Process

The message recovery process can be distilled into two key operations: the recovery of codewords within a sub-block, and successive interference cancellation across sub-blocks. To carry out this process, the destination seeks to start decoding with sub-blocks that are lightly loaded. Let  $R_j = |\mathcal{N}_j|$  denote the number of codewords transmitted during sub-block  $j$ . Based on its overall observations, the decoder can estimate  $R_j$  for  $j \in [1 : V]$ . A simple estimator can be obtained based on the received signal energy measured within a sub-block,

$$\hat{R}_j = \text{round} \left( \frac{\|\vec{y}_j\|^2 - N_s \sigma^2}{N_c P_c + N_p P_p} \right)$$

where  $\sigma^2$  is the noise variance, which is normalized to one throughout this article. This then yields an ordering for the early processing of sub-blocks, from lightly loaded to heavily loaded. We note that more sophisticated estimates based on decoding can be obtained, if necessary. Yet, empirical evidence suggests that the rudimentary energy estimator described above suffices for the task at hand.

Within a sub-block, the recovery process relies on a compressive sensing algorithm and a decoder for the  $T$ -user Gaussian multiple access (GMAC) channel. Once a message is acquired, interference cancellation is straightforward because the transmission schedule associated with a codeword is a deterministic function of its information bits. We will make this last statement precise shortly.

1) *Compressive Sensing Decoder*: As shown in (4), the received signal corresponding to a sub-block is equal to the sum of its sub-block codewords plus noise. Since the head of every sub-block codeword is a coded version of the preamble, the first  $N_p$  received symbols within sub-block  $j$  can be written as

$$\vec{y}_j[1 : N_p] = \sum_{i \in \mathcal{N}_j} \vec{a}_{w_p, i}^T + \vec{z}_j[1 : N_p]. \quad (7)$$

This expression is equivalent to the canonical compressive sensing matrix equation,

$$\vec{y}_j[1 : N_p]^T = \mathbf{A} \vec{v} + \vec{z}_j[1 : N_p]^T \quad (8)$$

where  $\vec{v} \in \{0, 1\}^{M_p}$  is a  $R_j$ -sparse vector that indicates the set of transmitted messages during sub-block  $j$ . We examine two approaches to recovering  $\vec{v}$  from  $\vec{y}_j[1 : N_p]$ . The first option is a simple correlation decoder, which is amenable to analysis. The second option exploits the sparsity of  $\vec{v}$  and the fact that the non-zero entries of  $\vec{v}$  are non-negative integers. The latter aspect makes this proposed receiver different from most standard compressed sensing reconstruction schemes. For sub-block  $j$ , we define the set of preamble message indices to be  $\mathcal{P}_j = \{w_p, i; i \in \mathcal{N}_j\}$ .

- *Correlation Decoder*: The preamble recovery process consists of computing the correlation between the received vector and the columns of the sensing matrix. The  $\hat{R}_j$  entries with the largest positive correlation values are declared the elements of  $\hat{\mathcal{P}}_j$ .
- *CS List Decoder*: Under this option, we first run a non-negative least squares or a non-negative,  $\ell_1$ -regularized LASSO algorithm that provides an estimate  $\tilde{v}$  of  $\vec{v}$ . Yet, these algorithms do not guarantee an output vector of the required sparsity or with elements contained in  $\{0, 1\}$ . To address this issue, we perform a hard thresholding operation on every element of  $\tilde{v}$  and we form a list of non-negative preamble indices  $\mathcal{P}_{\text{list}} = \{k : \tilde{v}_k > \eta\}$ . The value of parameter  $\eta$  is chosen such that the list size remains larger than  $T$ . We then implement a maximum likelihood decoder within the above list of indices to find the collection of  $T$  indices that best explain received vector  $\vec{y}_j[1 : N_p]$ ,

$$\hat{\mathcal{P}}_j = \arg \min_{\substack{\mathcal{P} \subseteq \mathcal{P}_{\text{list}} \\ |\mathcal{P}| \leq T}} \|\vec{y}_j[1 : N_p]^T - \sum_{k \in \mathcal{P}} \mathbf{A} \vec{e}_k\|_2^2 \quad (9)$$

where  $\{\vec{e}_k\}$  are the elements of the canonical basis for  $\mathbb{R}^{M_p}$ .

We emphasize that the size of  $\mathcal{P}_{\text{list}}$  grows as we decrease the value of threshold  $\eta$ , which in turn increases the complexity of the maximum likelihood estimator in (9). Conversely, if we raise the value of  $\eta$ , the list size decreases, complexity is reduced, and performance worsens. Thus, for a given SNR, the value of  $\eta$  should be optimized.

In both cases, the output of the compressive sensing algorithm for sub-block  $j$  is a set of preamble indices,  $\hat{\mathcal{P}}_j$ , where  $|\hat{\mathcal{P}}_j| \leq T$ . This collection of preamble sequences is communicated to the message-passing GMAC decoder. The residual error can be employed to assess the likelihood of a CS

decoding failure or a preamble collision. For instance, suppose two messages share a same preamble. In the absence of noise, even with the exact list  $\mathcal{P} = \text{set}(\mathcal{P}_j)$ , we have

$$\|\mathbf{A}\bar{v} - \sum_{k \in \mathcal{P}} \mathbf{A}\bar{e}_k\|_2^2 = N_p P_p$$

because elements cannot be duplicated in  $\mathcal{P}$ . When noise is present, the residual error

$$\left\| \bar{y}_j[1 : N_p]^T - \sum_{k \in \hat{\mathcal{P}}_j} \mathbf{A}\bar{e}_k \right\|_2^2 \quad (10)$$

becomes random. Nevertheless, its statistical properties can be employed to predict whether a preamble collision has occurred or not. This statistic can also be utilized to inform the successive interference cancellation step, starting with sub-blocks where the residual error is small whenever possible.

2) *Message Passing for Gaussian MAC*: We describe the structure of the joint decoding algorithm for the case where  $T = 2$ , yet this process can be generalized to larger values of  $T$  in a straightforward manner. Consider a generic sub-block with only two devices transmitting. For ease of exposition, suppose user 1 sends message  $w_1$  with representative sub-indices  $w_{p,1}$  and  $w_{c,1}$ . Likewise, assume user 2 transmits message  $w_2$  with sub-indices  $w_{p,2}$  and  $w_{c,2}$ . Owing to our two-step encoding process, we assume that preamble sub-indices  $\{w_{p,1}, w_{p,2}\}$  have already been recovered faithfully, and they are therefore available to the message passing decoder. After demodulation, the received signal pertaining to the (SC)LDPC codewords is given by

$$\bar{y}_j[N_p + 1 : N_s] = \pi_{\tau(w_{p,1})}(\bar{c}(w_{c,1})) + \pi_{\tau(w_{p,2})}(\bar{c}(w_{c,2})) + \bar{z}_j[N_p + 1 : N_s]. \quad (11)$$

The sequence of symbols  $\bar{y}_j[N_p + 1 : N_s]$  serves as input to the joint belief propagation (BP) decoder. As we can observe in (11), the coded symbols are subject to different permutations, each determined by preamble bits  $w_{c,i}$ . Thence, within the joint BP decoder, the effects of these permutations must be accounted for whenever messages are being sent to and from the MAC nodes. A schematic of the joint Tanner graph for the two-message case appears in Fig. 3.

Given received signal  $\bar{y}_j[N_p + 1 : N_s]$ , the joint BP decoder proceeds iteratively in a manner akin to a single-user AWGN channel decoder, aside from the extra step whereby messages are sent to and received from the MAC nodes in each iteration. Throughout this section, we use superscript to distinguish between messages 1 and 2. We also abuse notation slightly in that we reuse bound variables  $i$  and  $j$  to explain the decoding process, departing from their use hitherto. Context should prevent confusion, as the steps and notation are somewhat standard in treatments of iterative decoding.

- $u_{i,\text{MAC}}^1, u_{i,j}^1$ : Messages passed from the bit node corresponding to  $i$ th code bit of user 1 to the corresponding MAC node and SCLDPC check node  $j$ , respectively.
- $v_{j,i}^1$ : Message passed from SCLDPC check node  $j$  of user 1 to the bit node  $i$  of user 1.
- $v_{\text{MAC},i}^1$ : Message passed from  $i$ th MAC node to the connected bit node of user 1.

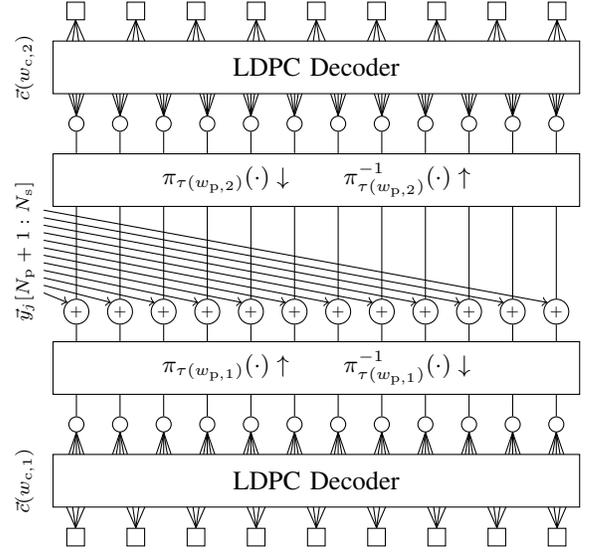


Fig. 3. This graph depicts the architecture for joint BP decoding. Symbols corresponding to the coded part of sub-block  $j$  are fed to the decoding algorithm. The effects of the permutations have to be accounted for when passing messages back and forth, as seen on the diagram. Beyond this distinction, decoding proceeds as expected.

The messages for user 2 are defined in an analogous manner. Figure 4 offers a visual synopsis of the message passing structure.

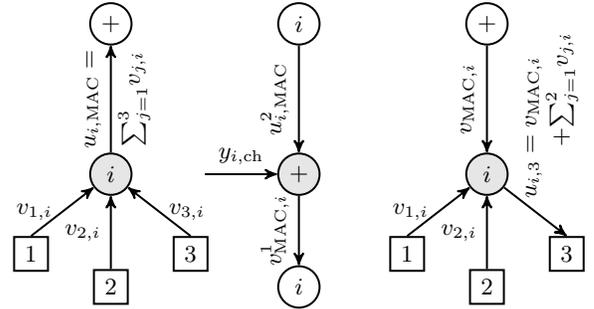


Fig. 4. Message passing rules at individual nodes on the joint Tanner graph of two users. The message passing rules at the check nodes of the SCLDPC code are identical to the single user channel coding case.

The message passing rules are summarized below.

- Bit node:

$$u_{i,j}^1 = v_{\text{MAC},i}^1 + \sum_{j' \in \mathcal{N}(i) \setminus j} v_{j',i}^1$$

$$v_{i,\text{MAC}}^1 = \sum_{j \in \mathcal{N}(i)} v_{j,i}^1.$$

- LDPC check node:

$$v_{j,i}^1 = 2 \tanh^{-1} \left( \prod_{i' \neq i} \tanh \left( \frac{u_{i',j}^1}{2} \right) \right).$$

- MAC node:

$$v_{\text{MAC},i}^1 = h(u_{i,\text{MAC}}^2, y_i)$$

$$v_{i,\text{MAC}}^2 = h(u_{i,\text{MAC}}^1, y_i) \quad (12)$$

where  $h(\ell, y|\sigma^2) = \log\left(\frac{1+e^\ell e^{2(y-1)/\sigma^2}}{e^\ell + e^{-2(y+1)/\sigma^2}}\right)$ .

Function  $h(\ell, y|\sigma^2)$  can be seen as the log-likelihood of variable  $x_2$  when  $y = x_1 + x_2 + z$ , with  $x_1, x_2 \in \{\pm 1\}$ , the log-likelihood ratio of variable  $x_1$  is known to be  $\ell$ , and  $z \sim \mathcal{N}(0, \sigma^2)$ .

For numerical simulations employing the message passing rules described above, we employ a collection SCLDPC codes for two-user GMAC channel. Performance for these codes is illustrated in Fig. 5.

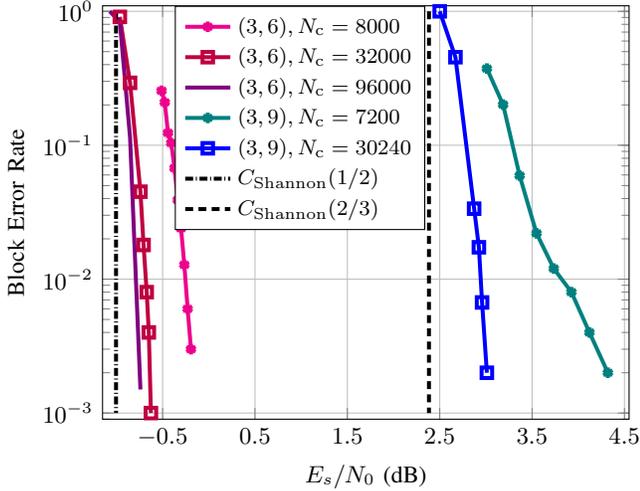


Fig. 5. We simulate the performance of the channel coding component alone using regular (3, 6) and (3, 9) spatially-coupled LDPC (SC-LDPC) ensembles for increasing blocklengths for two-user Gaussian MAC channel. The results demonstrate that it is possible to achieve the capacity of two-user GMAC (and can be generalized for  $T$ -GMAC) using identical codebooks at all the users, with independent permutations employed at each user similar to the encoding scheme described.

3) *Decoding Process Across Sub-Blocks (SIC)*: For any sub-block  $j$  with  $\hat{R}_j \leq T$ , the aforementioned process takes place. Upon successful decoding, the compressive sensing algorithm and the  $T$ -user GMAC decoder together output a set of transmitted messages within the designated sub-block. The recovered signals  $\{w_i : i \in \mathcal{N}_j\}$  are then inspected and their interference to other sub-blocks is removed. In particular, the sub-blocks where codeword  $w_i$  is repeated can be inferred through scheduling function  $g(\cdot)$ . The corresponding codeword  $\bar{x}(w_i)$  is subtracted, or peeled off, from the received signal  $\bar{y}_{j'}$  for all  $j' \in g(w_i) \setminus j$ . For every such sub-block, estimate  $\hat{R}_{j'}$  is updated to account for the subtraction of one interfering codeword. The above process is repeated until either all the  $K_a$  messages are decoded, or no sub-blocks with fewer than  $T$  codewords remain.

#### IV. ANALYSIS AND PARAMETER SELECTION

In this section, we look at parameter selection for the different components of the system, and we analyze how each one affects overall performance. As discussed in Section II, let  $\varepsilon$  denote the maximum per message error probability. That is, we wish to successfully decode a  $(1 - \varepsilon)$ -fraction of the  $K_a$  sent messages, on average. We introduce multiple error events to facilitate the characterization of the per message error

probability,  $P_e$ . Most of these events occur at the sub-block level, and they are described below for sub-block  $j$ .

- $\mathcal{E}_{p,j}$ : The event that, in the absence of a preamble collision, the compressive sensing algorithm fails to recover the set of preamble messages accurately,  $\hat{\mathcal{P}}_j \neq \mathcal{P}_j$ . In characterizing performance, we will introduce the related event  $\mathcal{E}_p$ , which is defined for the worst case  $R_j = T$ .
- $\mathcal{E}_{e,j}$ : This event pertains to error with respect to preamble collisions. Specifically, there are two types of error. A false positive or type-I error,  $\mathcal{E}_{e,j}^{(1)}$ , occurs when there are no preamble collisions and the CS algorithm successfully recovers the preambles, but the energy test based on (10) erroneously declares a failure. And, a false negative or type-II error,  $\mathcal{E}_{e,j}^{(2)}$ , corresponds to an undetected preamble collision. Since such events are mutually exclusive, we have  $\mathcal{E}_{e,j} = \mathcal{E}_{e,j}^{(1)} \cup \mathcal{E}_{e,j}^{(2)}$ .
- $\mathcal{E}_{c,j}$ : The event that the joint MAC channel decoder fails to recover the remaining messages when the number of such messages within sub-block  $j$  is no greater than  $T$ .
- $\mathcal{E}_{SIC}$ : The event that the message from an active device is not recovered by the successive interference cancellation process.

A decoding error within sub-block  $j$  can be expressed as  $\mathcal{E}_j = \mathcal{E}_{e,j} \cup \mathcal{E}_{p,j} \cup \mathcal{E}_{c,j}$ . Using the total probability theorem, we derive a bound on the per message error probability,  $P_e = \Pr(\mathcal{E}_{SIC})$ ,

$$\begin{aligned} P_e &\leq \Pr\left(\mathcal{E}_{SIC} \mid \left(\bigcup_j \mathcal{E}_j\right)^c\right) + \Pr\left(\bigcup_j \mathcal{E}_j\right) \\ &= \Pr\left(\mathcal{E}_{SIC} \mid \bigcap_j \mathcal{E}_j^c\right) + \Pr\left(\bigcup_j \mathcal{E}_{e,j} \cup \mathcal{E}_{p,j} \cup \mathcal{E}_{c,j}\right) \\ &\leq \Pr(\mathcal{E}_{SIC}^*) + \sum_j (\Pr(\mathcal{E}_{e,j}) + \Pr(\mathcal{E}_{p,j}) + \Pr(\mathcal{E}_{c,j})) \\ &\leq \Pr(\mathcal{E}_{SIC}^*) + V (\Pr(\mathcal{E}_e) + \Pr(\mathcal{E}_p) + \Pr(\mathcal{E}_c)) \end{aligned} \quad (13)$$

where  $\mathcal{E}_{SIC}^*$  is the event of a user not being recovered under the SIC decoder assuming that the compressed sensing decoder, the collision detector, and the channel decoder do not make any errors. This can be viewed as the error probability of an idealized successive interference cancellation process where the content of every sub-block with fewer than  $T$  messages is recovered faithfully, yet the iterative peeling process may fail. The multiplication factor  $V$  on the last line of (13) arises from applying the union bound across sub-blocks. The next step in our analysis is to derive bounds and approximations for the probabilities in (13).

##### A. Compressive Sensing Decoding

We begin our analysis of performance by establishing a ceiling on the probability of the compressive sensing algorithm failing to recover the preambles in the absence of collisions. This probability depends on system implementation. Recall the two reconstruction schemes introduced in Section III-B1, namely the Correlation Decoder and the CS List Decoder. We explore their performance for two classes of sensing matrices.

- *Random Ensemble*: The entries of the random sensing matrix are generated independently using a Rademacher distribution. That is, for  $\mathbf{A} \in \mathbb{R}^{N_p \times M_p}$ , every entry is

selected at random from  $\{\pm\sqrt{P_p}\}$  with equal probabilities.

- **Binary Ensemble:** A sensing matrix is derived as a subset of a binary linear code with appropriate scaling and shifting. For fixed  $N_p$  and  $M_p$ , we let  $\mathcal{C}_{\text{bin}}$  be a subset of size  $M_p$ , not necessarily a linear sub code, of a binary linear code with blocklength  $N_p$ . Then, the columns of sensing matrix are obtained via  $\sqrt{P_p}(\mathbf{1} - 2\mathbf{c})$  where  $\mathbf{c} \in \mathcal{C}_{\text{bin}}$ . We define  $d_{\min}$  and  $d_{\max}$  of the code  $\mathcal{C}_{\text{bin}}$  as follows

$$d_{\min} = \min_{c_1, c_2 \in \mathcal{C}_{\text{bin}}} d_H(c_1, c_2) \quad (14)$$

$$d_{\max} = \max_{c_1, c_2 \in \mathcal{C}_{\text{bin}}} d_H(c_1, c_2). \quad (15)$$

Clearly,  $d_{\min}$  and  $d_{\max}$  denote the minimum and maximum Hamming distances between any two binary vectors in  $\mathcal{C}_{\text{bin}}$  by  $d_{\min}$  and  $d_{\max}$ , respectively.

For these two matrix ensembles, the following lemma characterizes the error probability of the correlation decoder.

**Lemma 1** (Correlation Decoding). *Consider the  $T$ -sparse support recovery problem. The probability of error for the correlation decoder is given by*

$$\Pr(\mathcal{E}_p) = 1 - (1 - (M_p - T) \Pr(\mathcal{E}_{\text{corr}}))^T \leq T(M_p - T) \Pr(\mathcal{E}_{\text{corr}}) \quad (16)$$

where  $\Pr(\mathcal{E}_{\text{corr}})$  denotes the probability of a correlation error, which is defined by

$$\langle \vec{y}_j[1 : P_p], \vec{a}_k^T \rangle \leq \langle \vec{y}_j[1 : P_p], \vec{a}_{k'}^T \rangle \quad (17)$$

for some  $k \in \mathcal{P}_j$  and  $k' \notin \mathcal{P}_j$ . This probability can be upper bounded for the matrix ensembles under consideration,

- **Random Ensemble:**

$$\Pr(\mathcal{E}_{\text{corr}}) \leq \exp\left(-\frac{N_p P_p}{2(2 + (2T - 1)P_p)}\right) \quad (18)$$

- **Binary Ensemble:**

$$\Pr(\mathcal{E}_{\text{corr}}) \leq \exp\left(-\frac{P_p d_{\max}(1 - T(1 - d_{\min}/d_{\max}))^2}{2}\right) \quad (19)$$

where the randomness is over the matrix ensemble and the communication noise.

*Proof:* Let  $k \in \mathcal{P}_j$  be fixed. The correlation decoder includes  $k$  on its list of preambles  $\widehat{\mathcal{P}}_j$  whenever  $\langle \vec{y}_j[1 : P_p], \vec{a}_k^T \rangle$  exceeds  $\langle \vec{y}_j[1 : P_p], \vec{a}_{k'}^T \rangle$  for all  $k' \notin \mathcal{P}$ . Since there are  $(M_p - T)$  candidates for  $k'$ , the probability that  $k \in \widehat{\mathcal{P}}_j$  becomes  $1 - (M_p - T) \Pr(\mathcal{E}_{\text{corr}})$ . For the overall process to be successful, this must occur  $T$  times and, hence, the probability of failure is given by (17). The analysis for event  $\mathcal{E}_{\text{corr}}$  and for the corresponding bounds in (18) and (19) are lengthier; they are provided in Appendix A. ■

For the random ensemble, we observe from (16) and (18) that the error probability for the correlation decoder decays exponentially in the number of channel uses  $N_p$ . Still, the

decay rate is very slow with respect to SNR  $P_p$ . In fact, performance converges to a non-zero constant

$$\lim_{P_p \uparrow \infty} \Pr(\mathcal{E}_{\text{corr}}) = \exp\left(-\frac{N_p}{2(2T - 1)}\right).$$

If we consider the binary ensemble, the error probability of the decoder decays exponentially in both the number of symbols  $N_p$  and SNR  $P_p$  provided that subset  $\mathcal{C}_{\text{bin}}$  possesses the following properties: the gap between the minimum and maximum Hamming distances,  $d_{\max} - d_{\min}$ , is small; the minimum distance  $d_{\min}$  is large. Together, these two conditions imply a large  $d_{\max}$ , which is necessary for a large exponent (see (19)). Based on the design objectives outlined above, we create a sensing matrix from the *binary ensemble* for an example with parameters  $M_p = 512$  and  $N_p = 63$ .

**Example 2** (Sensing Matrix from Binary Code). *Consider a binary BCH code  $\mathcal{C}_{\text{BCH}}$  with parameters  $(n, k) = (63, 10)$  and size 1024. Let subset  $\mathcal{C}_{\text{bin}} \subset \mathcal{C}_{\text{BCH}}$  correspond to a partition where  $\mathbf{c} \in \mathcal{C}_{\text{bin}}$  if and only if  $\mathbf{1} \oplus \mathbf{c} \in \mathcal{C}_{\text{BCH}} \setminus \mathcal{C}_{\text{bin}}$ ;  $\mathbf{1} \oplus \mathbf{c}$  is the one's complement of  $\mathbf{c}$ , and  $\mathbf{0} \in \mathcal{C}_{\text{bin}}$ . For this decomposition, the weight and distance parameters of  $\mathcal{C}_{\text{bin}}$  are computed to be  $(w_{\min}, w_{\max}, d_{\min}) = (27, 36, 27)$  where  $w_{\min}, w_{\max}$  are the minimum and maximum Hamming weights of binary vectors in  $\mathcal{C}_{\text{bin}}$ . We build sensing matrix  $\mathbf{A}$  of dimension  $N_p \times M_p = 63 \times 512$  by selecting columns of the form  $\vec{a}_k = \sqrt{P_p}(\mathbf{1} - 2\mathbf{c}_k)$  where  $\mathbf{c}_k \in \mathcal{C}_{\text{bin}}$ . This decomposition enables us to maintain minimum distance  $d_{\min} = 27$ , while reducing the maximum distance for the subset  $\mathcal{C}_{\text{bin}}$  to  $d_{\max} = 36$  from 63 of the original code.*

While the performance of LASSO and non-negative least squares is significantly better than that of the correlation decoder, their performances are difficult to characterize analytically. This precludes the derivation of meaningful bounds on the performance of the CS list decoder. Nevertheless, empirical evidence suggests that sensing matrices designed based on BCH codes perform very well even with the CS list decoder. Figure 6 showcases error performance results for preamble recovery. The code-based matrix construction is that of Example 2.

## B. Errors and Preamble Collisions

We turn to error events that are associated with preamble collisions,  $\mathcal{E}_{e,j}$ . As mentioned at the onset of this section, there are two types of error, which we analyze separately. As a preliminary step, we look at the probability that a preamble collision occurs.

**Lemma 3.** *In any sub-block with  $T$  or fewer transmissions, the probability that at least two messages within a sub-block share a same preamble is bounded by*

$$\Pr(\mathcal{E}_{\text{coll}}) \leq 1 - \prod_{i=0}^{T-1} \left(1 - \frac{i}{M_p}\right) \leq \frac{T(T-1)}{2M_p}. \quad (20)$$

*Proof:* We consider the event  $\mathcal{E}_{\text{coll},j}^c$  in which the  $R_j$  messages in sub-block  $j$  have unique preamble indices. There

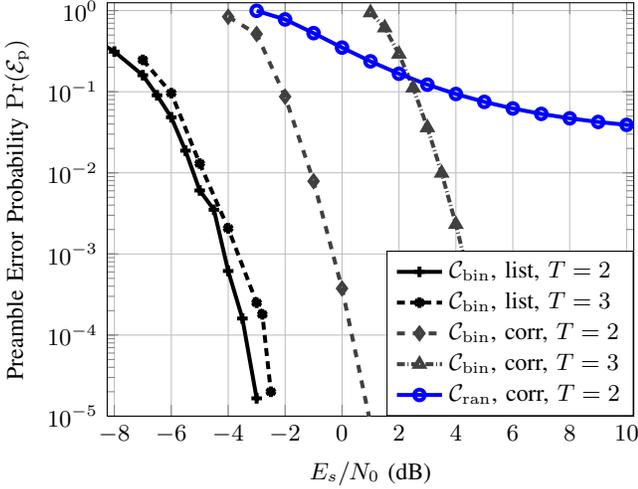


Fig. 6. This plot compares the performance of the binary and random matrix ensembles under CS list decoding and correlation decoding. The parameters used in the figure are  $N_p = 63$ ,  $M_p = 512$ , and  $T \in \{2, 3\}$ . The sensing matrix for the binary ensemble is given in Example 2. For the correlation decoder, the graph reports the performance bounds given in Lemma 1; whereas for list decoding, results are based on numerical simulations with a non-negative least squares followed by maximum likelihood as a CS sub-component.

are a total of  $M_p$  possible preamble indices. When they are equally likely, we get

$$\Pr(\mathcal{E}_{\text{coll},j}^c) = \frac{M_p(M_p - 1) \cdots (M_p - (R_j - 1))}{M_p^{R_j}}. \quad (21)$$

This then implies

$$\Pr(\mathcal{E}_{\text{coll},j}) = 1 - \prod_{i=0}^{R_j-1} \left(1 - \frac{i}{M_p}\right) \leq \frac{R_j(R_j - 1)}{2M_p}, \quad (22)$$

where the last term is a bound derived from the first-order Taylor approximation of  $\Pr(\mathcal{E}_{\text{coll},j})$ . According to the encoding scheme, every message is randomly assigned to sub-blocks based on the message index and, hence, the degree distribution for the occupation of sub-blocks is non-uniform. Still, we observe that the CS algorithm and the channel decoder are run on a sub-block only when the remaining number of signals are less than  $T$ . Thus, we can set the number of messages per sub-block to  $T$  to get a uniform upper bound across the sub-blocks. Substituting  $R_j$  with  $T$  in (22), we obtain the desired expression. In the decoding process, only sub-blocks with  $T$  or fewer transmissions are decoded and, hence, it suffices to consider the probability of two messages picking the same preamble in sub-blocks with  $T$  or fewer transmissions. ■

Again, error event  $\mathcal{E}_{e,j}^{(1)}$  takes place when there are no preamble collisions and the CS algorithm recovers the preambles, but the energy test based on (10) erroneously declares a failure. The probability of this event is bounded in Lemma 4.

**Lemma 4.** Suppose that the energy test of (10) utilizes a threshold of  $\xi > 0$ , then

$$\Pr(\mathcal{E}_{e,j}^{(1)}) \leq (\xi e^{1-\xi})^{\frac{N_p}{2}}.$$

*Proof:* If there are no preamble collisions and the compressed sensing decoder successfully recovers the indices, then

$$\begin{aligned} \Pr(\mathcal{E}_{e,j}^{(1)}) &= \Pr\left(\left\|\vec{y}_j[1 : N_p]^T - \sum_{k \in \mathcal{P}_j} \mathbf{A}\vec{e}_k\right\|_2^2 > \xi\right) \\ &= \Pr\left(\frac{1}{N_p} \left\|\vec{z}_j[1 : N_p]^T\right\|_2^2 > \xi\right) \\ &\leq (\xi e^{1-\xi})^{\frac{N_p}{2}} \end{aligned}$$

where the last inequality is a Chernoff bound on the tail probability of a chi-squared distribution. ■

A false negative,  $\mathcal{E}_{e,j}^{(2)}$ , arises from an undetected preamble collision. The probability of such an event is examined below.

**Lemma 5.** Suppose that the energy test of (10) utilizes a threshold of  $\xi > 0$ , then

$$\Pr(\mathcal{E}_{e,j}^{(2)}) \leq 1 - \mathcal{Q}_{\frac{N_p}{2}}\left(\sqrt{N_p P_p}, \sqrt{N_p \xi}\right)$$

where  $\mathcal{Q}_M(\cdot, \cdot)$  is the Marcum  $\mathcal{Q}$ -function.

*Proof:* The hardest preamble collision event to detect is when exactly two messages share a same preamble. Denote the duplicate preamble by  $k'$ . Then, the repeated entry will not be accounted for in  $\mathcal{P}_j$ , and

$$\vec{y}_j[1 : N_p]^T - \sum_{k \in \mathcal{P}_j} \mathbf{A}\vec{e}_k = \vec{z}_j[1 : N_p]^T + \mathbf{A}\vec{e}_{k'}.$$

This statistic has noncentral chi-squared distribution, with  $N_p$  degrees of freedom and noncentrality parameter  $\|\mathbf{A}\vec{e}_{k'}\|^2$ . For the two matrix ensembles considered herein, the noncentrality parameter becomes  $N_p P_p$ . Then, we get

$$\begin{aligned} \Pr(\mathcal{E}_{e,j}^{(2)}) &= \Pr\left(\frac{1}{N_p} \left\|\vec{z}_j[1 : N_p]^T + \mathbf{A}\vec{e}_{k'}\right\|_2^2 \leq \xi\right) \\ &\leq 1 - \mathcal{Q}_{\frac{N_p}{2}}\left(\sqrt{N_p P_p}, \sqrt{N_p \xi}\right) \end{aligned}$$

where  $\mathcal{Q}_M(\cdot, \cdot)$  is the Marcum  $\mathcal{Q}$ -function. ■

After these preliminaries, we are ready to address the probability of event related to preamble collisions.

**Proposition 6.** For the energy test of (10) with threshold of  $\xi > 0$  and the matrix ensembles at hand, the probability of  $\mathcal{E}_e$  is bounded by

$$\begin{aligned} \Pr(\mathcal{E}_e) &\leq (\xi e^{1-\xi})^{\frac{N_p}{2}} \\ &\quad + \frac{T(T-1)}{2M_p} \left(1 - \mathcal{Q}_{\frac{N_p}{2}}\left(\sqrt{N_p P_p}, \sqrt{N_p \xi}\right)\right) \end{aligned}$$

where  $\mathcal{Q}_M(\cdot, \cdot)$  is the Marcum  $\mathcal{Q}$ -function.

*Proof:* To establish this result, we condition on  $\mathcal{E}_{\text{coll}}$  to obtain

$$\begin{aligned} \Pr(\mathcal{E}_e) &= \Pr(\mathcal{E}_e | \mathcal{E}_{\text{coll}}^c) \Pr(\mathcal{E}_{\text{coll}}^c) + \Pr(\mathcal{E}_e | \mathcal{E}_{\text{coll}}) \Pr(\mathcal{E}_{\text{coll}}) \\ &\leq \Pr(\mathcal{E}_e | \mathcal{E}_{\text{coll}}^c) + \Pr(\mathcal{E}_e | \mathcal{E}_{\text{coll}}) \Pr(\mathcal{E}_{\text{coll}}) \\ &= \Pr(\mathcal{E}_e^{(1)}) + \Pr(\mathcal{E}_e^{(2)}) \Pr(\mathcal{E}_{\text{coll}}). \end{aligned}$$

The statement of the proposition is obtained by substituting the above expression from the corresponding upper bounds in Lemma 4, Lemma 5, and Lemma 3. Parameter  $\xi$  can be selected as to optimize the bound. ■

### C. Channel Coding Problem

This section focuses on the analysis of the  $T$ -GMAC channel coding problem. Although information theoretic limits for the multiple access problem and, in particular, the symmetric rate region are well known, these results offer only limited insight into coding with short blocklengths. Since we are applying coding over short sub-blocks, we turn to finite length performance. In this context, the finite-length random coding bounds derived by Polyanskiy [1] for the Gaussian multiple access channel are more revealing. The following lemma is equivalent to [1, Theorem 1], except that we are interested in the situation where an error is declared whenever at least one of the messages fails to appear in the decoded set (see event  $\mathcal{E}_{c,j}$ ). This contrasts with [1], where error events and their probabilities are defined in a manner akin to (2).

**Lemma 7.** *There exists an  $(N_c, M_c)$  random-access code for  $T$ -user satisfying the power constraint  $P$ , as in (5), with probability of error under maximum-likelihood decoder bounded by*

$$P(\mathcal{E}_c) \leq \mathfrak{h}_{\text{FBL}}(N_c, M_c, T, P) = p_0 + \sum_{t=1}^T \min(p_t, q_t) \quad (23)$$

where

$$p_0 = \frac{\binom{T}{2}}{M_c} + T \Pr\left(\sum_{j=1}^{N_c} z_j^2 > \frac{N_c P}{P'}\right)$$

$$p_t = \exp(-N_c E(t))$$

$$q_t = \inf_{\gamma} (\Pr(I_t \leq \gamma) + \exp(N_c(R_1 + R_2) - \gamma))$$

$$E(t) = \max_{0 \leq \rho, \rho_1 \leq 1} (E_0(\rho, \rho_1) - \rho \rho_1 t R_1 - \rho_1 R_2)$$

$$E_0 = \rho_1 a + \frac{1}{2} \log(1 - 2b\rho_1)$$

$$a = \frac{\rho}{2} \log(1 + 2P't\lambda) + \frac{1}{2} \log(1 + 2P't\mu)$$

$$b = \rho\lambda - \frac{\mu}{1 + 2P't\mu}$$

$$\mu = \frac{\rho\lambda}{1 + 2P't\lambda} \quad \lambda = \frac{P't - 1 + \sqrt{D}}{4(1 + \rho_1\rho)P't}$$

$$D = (P't - 1)^2 + 4P't \frac{1 + \rho\rho_1}{1 + \rho}$$

$$R_1 = \frac{1}{N_c} \log M_c - \frac{1}{N_c} \log(t!) \quad R_2 = \frac{1}{N_c} \log \binom{T}{t}.$$

In addition,  $\vec{z} \sim \mathcal{N}(0, \mathbf{I}_{N_c})$  is white Gaussian noise, and

$$I_t = \min_{\substack{|S_0|=t \\ S_0 \subseteq [1:T]}} N_c C_t + \frac{\log e}{2} \left( \frac{\|\sum_{i \in S_0} \vec{c}_i + \vec{z}\|_2^2}{1 + P't} - \|\vec{z}\|_2^2 \right)$$

$$C_t = \frac{1}{2} \log(1 + P't).$$

*Proof:* In [1], the author considers the  $T$ -user GMAC problem with a power constraint  $P$ , akin to (5). To explain their results, let  $W$  be the set of messages of size  $T$ , chosen by the users uniformly without replacement. Also, let  $\widehat{W}$  be the collection of messages of size  $T$  output by the de-

coder. Consider a random Gaussian codebook with distribution  $\mathcal{N}(0, P'\mathbf{I}_n)$ , with  $P' < P$ . The author in [1] shows that

$$\Pr\left(|W \setminus \widehat{W}| = t\right) = \min(p_t, q_t). \quad (24)$$

They also prove that  $p_0$  is the total variation distance of a random variable of maximum value 1 under the following two conditions. Messages are sampled independently, rather than without replacement. And, a codeword is set to  $\mathbf{0}$  whenever the total power of its candidate random vector exceeds  $N_c P$ . These findings, along with the observation that

$$\Pr(\mathcal{E}_c) = p_0 + 1 - \prod_{i=1}^t \left(1 - \Pr(\hat{w}_i \notin \widehat{W})\right)$$

$$\leq p_0 + \sum_{t=1}^T \Pr(|S \setminus \hat{S}| = t),$$

complete the proof.  $\blacksquare$

### D. Successive Interference Cancellation

The channel coding literature on LDPC codes applied to binary erasure channels, and the literature on sparse signal recovery via message passing over Tanner graphs are well developed [16], [10]. In these situations, successive interference cancellation is often studied under the peeling decoder designation. This refers to the iterative process where a right node (time sub-block) connected to only one left node (message) can be decoded and removed from all its connections, thereby peeling off the node from the bipartite graph. This concept essentially coincides with the successive interference cancellation process described in Section II, except that our peeling process can proceed as long as the number of variable nodes connected to a sub-block is less than or equal to  $T$  instead of 1. Although density evolution is well studied to predict the performance of such a decoding process for the base case, existing density evolution methods are essentially constrained to the scenario where  $T = 1$ . Before we address this issue, we describe the enhanced peeling process formally.

**Definition 8** (Idealized SIC Decoding). *An idealized SIC decoder is a decoder for which all users within a sub-block can be decoded with zero error whenever the number of residual undecoded users in the sub-block is less than or equal to  $T$ . If the number of undecoded users within a sub-block exceeds  $T$ , then the corresponding messages cannot be recovered from this sub-block at this stage. In other words, under this idealized SIC process, there are no preamble collisions in any sub-block. Moreover, the channel and sparse signal decoders are assumed reliable, with zero error. This peeling process continues iteratively until all the users are decoded or all the remaining sub-blocks features more than  $T$  undecoded users. We also refer to this as the idealized BP process.*

In this section, we address this gap by analyzing the density evolution for the idealized SIC decoding and deriving performance thresholds. See [16] for additional information on the threshold behavior of iterative peeling processes. Here again, we abuse notation slightly in that we reuse variables  $x$  and  $y$  to discuss density evolution. This should not be

an issue to the reader as this notation is widely used in the study of message passing, and context should prevent possible confusion.

**Lemma 9** (Density Evolution). *Let the left and right degree distributions of the bipartite graph from the edge perspective be  $\lambda(x)$  and  $\rho(x)$ . Also, let  $x_t$  be the probability that an edge in the graph, in iteration  $t$  of the idealized BP process, is connected to a left node that is undecoded yet. Then, the recurrence relation for  $x_t$  corresponding to the idealized BP process is given by*

$$y_t = \sum_{r=1}^T \rho_r + \sum_{r>T} \rho_r \left( \sum_{t=0}^{T-1} \binom{r-1}{t} (1-x_t)^{r-1-t} x_t^t \right) \quad (25)$$

$$x_{t+1} = \lambda(1-y_t). \quad (26)$$

*Proof:* For an LDPC code, the bipartite graph corresponds to the parity check matrix where the left and right nodes represent the bits of the codeword and the parity check equations, respectively. If we consider an LDPC code under a binary erasure channel where each bit is erased with probability  $\varepsilon$ , under the assumption that the bipartite graph is a tree, the probability that a random edge in the graph is an erasure in iteration  $t$  of the peeling process is given by [16]

$$y_t = \sum_{r=1}^{r_{\max}} \rho_r (1-x_t)^{r-1} \quad (27)$$

$$x_{t+1} = \varepsilon \lambda(1-y_t). \quad (28)$$

Equation (27) stems from the observation that the incoming messages at a check node are independent, due to the tree structure; and, the outgoing message on an edge from a check node of degree  $r$  is a non-erasure if and only if all the incoming messages are non-erasures. For degree distributions with finite maximum degrees on the left and right, it is known that a graph chosen randomly from the ensemble  $(N, \lambda, \rho)$  is a tree with probability approaching one, asymptotically in the blocklength of the code.

If we consider edge  $e$  connected to a check node of degree  $r$  in the idealized BP process, the outgoing message is a non-erasure if and only if there are at most  $T-1$  erasures in the remaining  $r-1$  incoming edges. Thus, the probability that the outgoing message from a check node of degree  $r$  forms a non-erasure, denoted by  $y_{t,r}$ , if every incoming message on the remaining  $r-1$  edges are erased with probability  $x_t$  is equal to

$$y_{t,r} = \begin{cases} \sum_{t=0}^{T-1} \binom{r-1}{t} (1-x_t)^{r-1-t} x_t^t & r > T \\ 1 & r \leq T. \end{cases}$$

Averaging over all edges, where an edge is connected to a check node of degree  $r$  with probability  $\rho_r$ , yields (25). ■

Let  $L(x) = \sum_{k=1}^{\ell_{\max}} L_k x^k$  be the left degree distribution with which each user chooses their repetition parameter, as described in Section III-A; that is,  $\Pr(\ell_w = k) = L_k$ . Also, let the average left degree of this distribution be  $\ell_{\text{avg}} = \sum_{k=1}^{\ell_{\max}} k L_k$ . Then, according to our transmission policy, the

right degree distribution  $R(x)$  is binomial with parameters  $(K_a \ell_{\text{avg}}, 1/V)$ . In the limit as  $K_a \rightarrow \infty$ ,  $R(x)$  converges to a Poisson distribution with parameter  $r_{\text{avg}} = \frac{K_a \ell_{\text{avg}}}{V}$ . Thus, asymptotically in  $K_a$ , we have  $R(x) = e^{-r_{\text{avg}}(1-x)}$  and  $\rho(x) = R'(x)/R'(1) = e^{-r_{\text{avg}}(1-x)}$ . For additional details, we refer the reader to [10], [18].

**Corollary 10.** *For  $V = \alpha K_a$ , where  $\alpha$  is fixed, the asymptotic performance of our transmission scheme under the idealized SIC decoding process can be characterized by*

$$\lim_{K_a \rightarrow \infty} \Pr(\mathcal{E}_{\text{SIC}}(K_a, T)) = L(1-y_\infty)$$

where  $y_\infty = \lim_{t \rightarrow \infty} y_t$  and  $\Pr(\mathcal{E}_{\text{SIC}}(K_a, T))$  denotes the probability that the idealized SIC process fails to recover a message when  $K_a$  devices are active. The initial condition is  $x_0 = 1$ ; variables  $x_t$  and  $y_t$  are governed by the density evolution equations in Lemma 9.

As we infer from (26) and (25),  $x_t = 0$  is a fixed point if and only if  $\lambda_0 = 0$ . This leads us to the following result characterizing the threshold behavior of the system.

**Definition 11** (Density Evolution Threshold). *For cases where  $L_1 = 0$ , we define the density evolution threshold  $\alpha_{\text{DE}}^*$  by*

$$\alpha_{\text{DE}}^* = \inf \left\{ \alpha : \lim_{K_a \rightarrow \infty} \Pr(\mathcal{E}_{\text{SIC}}(K_a, T)) = 0 \right\}.$$

This definition is motivated by similar concepts in the literature. We validate the threshold behavior via numerical simulations. For a fixed left degree distribution  $L(x) = x^2$ , we first compute the density evolution thresholds according to Definition 11. This produces  $\alpha_{\text{DE}}^* = 0.5975$  and  $0.2949$  for  $T \in \{2, 4\}$ , respectively. We then perform Monte Carlo simulations wherein a random graph is generated every time, as described in Section III-A, for increasing values of  $K_a$ . Figure 7 plots performance as functions of the number of sub-blocks. In both cases, simulations offer supporting empirical evidence of the threshold behavior. As  $K_a$  increases, the probability of a user not being decoded decreases sharply for values of  $\alpha > \alpha_{\text{DE}}^*$ ; yet, this probability remains fairly constant for values of  $\alpha \leq \alpha_{\text{DE}}^*$ . As can be observed from Fig. 7, for finite values of  $K_a$  and given a desired upper bound on  $\Pr(\mathcal{E}_{\text{SIC}}(K_a, T))$ ,  $\alpha_{\text{DE}}^*$  serves as a good starting point in the search for required value of  $\alpha = \frac{V}{K_a}$ . A precise characterization of error probability,  $\Pr(\mathcal{E}_{\text{SIC}}(K_a, T))$ , for idealized SIC decoder for  $T > 1$ , is an interesting future direction.

## V. NUMERICAL RESULTS

In this section, we assess the overall performance of our proposed framework and we compare it with the performance of other schemes found in the literature. In [5], Ordentlich and Polyanskiy propose a low-complexity coding scheme for the uncoordinated, uncoordinated Gaussian multiple access channel. They also contrast their low complexity coding scheme with slotted ALOHA, treating inference as noise (TIN), and random coding achievability bounds. To ensure fairness in comparison, we pick identical parameters in our simulations. That is, we fix the number of information bits per message to  $B = 100$ ;

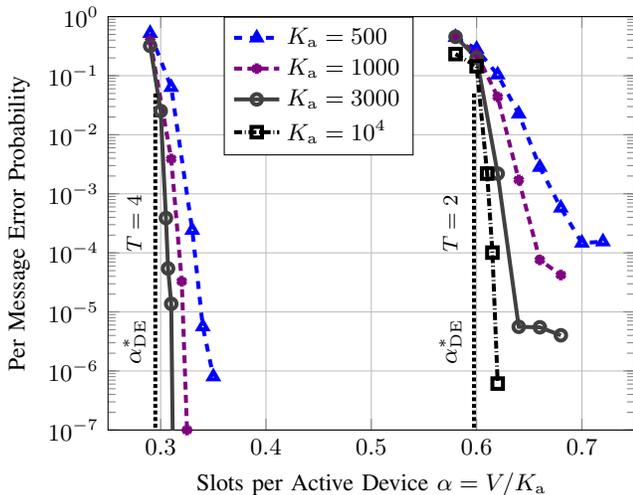


Fig. 7. In this plot,  $\alpha_{\text{DE}}^*$  is the density evolution threshold computed for  $L(x) = x^2$  and  $T = \{2, 4\}$  from Lemma 9. We validate the threshold behavior by evaluating the performance of the idealized BP process via Monte Carlo simulations for increasing blocklengths. We observe that the simulations indeed confirm the threshold behavior for values of  $\alpha$  above the DE threshold.

the total number of channel uses is set to  $N = 30,000$ ; the number of active users is  $K_a \in [25 : 300]$ ; and the maximum per user error probability is  $P_e \leq \varepsilon = 0.05$ .

With these common parameters fixed, we turn to the design parameters that are specific to our proposed scheme. We explore performance for systems where the maximum number of messages to be jointly decoded falls within  $T \in \{2, 4, 5\}$ . The left degree distribution is chosen to be  $L(x) = \beta x + (1 - \beta)x^2$  (see Remark 12), and the free parameter is optimized over  $\beta \in \{0, 0.1, \dots, 1\}$ . The number of preamble bits is  $B_p = 9$ , and the channel code operates over  $B_c = 91$  bits. Sensing matrix  $\mathbf{A} \in \mathbb{R}^{63 \times 512}$  is as described in Example 2. We briefly explain the rationale behind our choices for  $B_p$  and  $\mathbf{A}$  in Remark 13.

The number of channel uses available for channel coding,  $N_c$ , depends on the sub-block size, which in turn is contingent on the total number of sub-blocks  $V$ . It would be impractical to build a channel code for all rates  $B_c/N_c$  (with  $B_c$  fixed and  $N_c$  varying), and subsequently evaluate performance numerically for every pair  $(B_c, N_c)$ . Consequently, we employ the finite blocklength achievability bound in (24) as a proxy for the operation of the channel code while evaluating overall system performance. For a specific operating point, we also report findings based on LDPC code constructions and compare with using finite blocklength achievability bounds. This appears to be a reasonable strategy.

To meet error probability constraint  $\varepsilon$ , we leverage (13) and aim at  $\Pr(\mathcal{E}_{\text{SIC}^*}) + V(\Pr(\mathcal{E}_p) + \Pr(\mathcal{E}_e) + \Pr(\mathcal{E}_c)) \leq 0.05$ . We further divide the probability of error as follows:  $\Pr(\mathcal{E}_p), \Pr(\mathcal{E}_e), \Pr(\mathcal{E}_c) \leq 0.01/3V$  and  $\Pr(\mathcal{E}_{\text{SIC}^*}) \leq 0.04$ . For a given  $T$ , the performance of the overall scheme is presented as the minimum  $E_b/N_0$  required to achieve  $P_e \leq \varepsilon$ , where

$$\frac{E_b}{N_0} = \min_{\beta} \frac{(2 - \beta)(N_p P_p + N_c P_c)}{2B} \quad (29)$$

$$P_p = \arg \min \left\{ P'_p : \max(\Pr(\mathcal{E}_p), \Pr(\mathcal{E}_e)) \leq \frac{0.01}{3V} \right\} \quad (30)$$

$$P_c = \arg \min \left\{ P'_c : h_{\text{FBL}}(N_c, B_c, T, P'_c) \leq \frac{0.01}{3V} \right\} \quad (31)$$

$$N_s = \lfloor N/V \rfloor \quad (32)$$

$$V = \arg \min \{ V' : \Pr(\mathcal{E}_{\text{SIC}^*}(K_a, V', T)) \leq 0.04 \}. \quad (33)$$

We compute  $\frac{E_b}{N_0}$  in (29) by computing Eqns. (33), (30), (32), and (31) in that order. A few remarks on how we compute these quantities are apropos. To evaluate  $\Pr(\mathcal{E}_p)$ , we choose  $T$  preamble message indices at random, without replacement, from the  $M_p$  admissible indices and we form the measurement vector. We then apply the list decoder described in Section III-B1. The probability of error  $\Pr(\mathcal{E}_p)$  in (30) is computed from at least  $10^5$  Monte Carlo samples. We use the upper bound for  $\Pr(\mathcal{E}_e)$  given in Proposition 6. Finally, to evaluate  $\Pr(\mathcal{E}_{\text{SIC}^*}(K_a, V', T))$ , we rely on Monte Carlo simulations. For each sample, we generate a bipartite graph with  $K_a$  variable nodes and  $V'$  sub-block nodes. Edge connections are as described in Section III-A. We then run the idealized SIC decoder on this bipartite graph. Empirical averaging serves as an estimate for the error probability. The results for the minimum SNR required to achieve the target error probability are presented in Fig. 8.

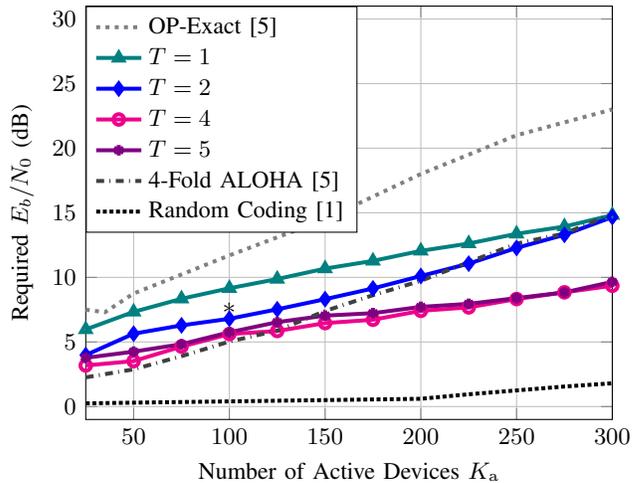


Fig. 8. This shows the minimum  $E_b/N_0$  required to achieve  $P_e \leq 0.05$  as a function of the number of active devices. The \* mark represents the performance of our proposed scheme ( $T = 2$ ) with a regular LDPC code, instead of the finite blocklength bounds given in [1]. The proposed framework significantly outperforms existing low-complexity, uncoordinated schemes for large systems.

In Fig. 8, the curves labeled  $T = 2$ ,  $T = 4$ , and  $T = 5$  correspond to the performance of our proposed scheme, evaluated as described above. The curve labeled 4-fold ALOHA is the performance of the 4-fold ALOHA scheme from [5] wherein the traditional slotted-ALOHA scheme is extended to the case where all the sub-blocks with number of interfering signals not greater than 4 are assumed to be decoded with finite achievability performance of 4-GMAC. It can be seen that, for large values of  $K_a$ , our proposed scheme with  $T \in \{4, 5\}$  substantially outperforms 4-fold ALOHA, despite the fact that the latter scheme necessitates acknowledgement feedback.

This gain is attributable to the iterative peeling process, which is absent in 4-fold ALOHA. The curve labeled OP-Exact is a reproduction of the findings from [5] for the practical scheme introduced therein. The \* mark points to the performance of our proposed scheme ( $T = 2$ ), when an LDPC code is simulated, instead of using the FBL bound as a performance proxy. Specifically, we employ a rate-1/4 (364, 91) LDPC code obtained from repeating every coded bit of a (3, 6) LDPC code twice. Decoding takes place under a joint message passing decoder. It can be seen that the simulation results for the (3, 6) LDPC code are only 0.5 dB away from the curve corresponding to  $T = 2$ , which offers supporting evidence to the approximation based on the FBL sub-block coding bound. More importantly, the figure shows that our proposed scheme offers substantial gain over the results in [5].

In the proposed encoding scheme, for  $L(x) = \beta x + (1 - \beta)x^2$ , each device may transmit once or twice depending on its message index. We emphasize that the power constraint employed is averaged over all possible message indices, i.e.,

$$\mathbb{E}_w [\|\tilde{c}(w)\|_2^2] = (2 - \beta)P.$$

For completeness, Fig. 9 presents similar results when the power constraint is uniform across all codewords,  $\|\tilde{c}(w)\|_2^2 \leq P$  for all  $w$ . We enforce this stricter constraint, for each value of SNR  $E_s/N_0$ , by choosing  $\beta = 0$  (or 1), which guarantees that every codeword is repeated exactly twice (or once) irrespective of the underlying message. As seen on the figure, performance deteriorates only slightly, which is encouraging.

**Remark 12.** We remark in Section IV-D that, if the minimum left degree is one, then zero is not a fixed point for the DE equations; in other words, in the asymptotic regime, error floors rather than threshold behavior will dominate. Still, the effects of a minimum left degree of one for a finite number of messages regime are not very clear.

**Remark 13.** We discuss the choice of compressive sensing parameters  $B_p, N_p$  &  $\mathbf{A}$  used in Sec. V.

- We observe from Fig. 6 that under practical decoding schemes the random ensemble performs worse than the carefully designed BCH-code based sensing matrix. Therefore, we choose BCH-code based designs for  $\mathbf{A}$  with  $N_p = 2^m - 1$
- At one end of the operating regime, for  $K_a = 300$ ,  $\alpha \approx 1.2$ , and  $T = 1$ , Eqn. (32) implies  $N_s = 83$ . We note that  $N_s = N_p + N_c$ .
- For a fixed  $N_p$ , we look at the choice of  $B_p$ . Increasing  $B_p$  by one bit results in a compressed sensing matrix with double the number of columns. The non-negative least squares method compressive sensing decoder has a complexity of  $O(M_p^3)$ . Thus an increase in  $B_p$  by one bit results in the computational complexity of compressed sensing decoder octupled. This makes the evaluation of Eqn. (30) for large sensing matrices, for various values of  $V$ , extremely difficult.

In view of the above constraints and the specific parameters of interest i.e.,  $N = 30000, \varepsilon = 0.05$  considered in this paper, a

good balance between error performance and complexity was obtained by choosing  $\mathbf{A}$  in Example 2 with size  $N_p \times M_p = 63 \times 512$ .

## VI. DISCUSSION

This article presents a new low-complexity framework for the unsourced, uncoordinated MAC channel. This novel framework employs the fact that the randomness contained in prospective messages can be leveraged to informed random access. It creates a scheme where transmissions are device agnostic, relying solely on content to create signals. Furthermore, it takes advantage of compressive sensing and joint decoding to enable successive interference cancellation. The resulting scheme outperforms other existing low-complexity schemes found in the literature.

Conceptually, the article emphasizes the connection between compressive sensing and the unsourced MAC. The proposed framework also invites new paradigms that connect sparse recovery with modern coding theory. In particular, it suggests new means to encode sparse signals for subsequent approximate reconstruction when signal spaces are vast. Possible avenues of future research include the implementation of this framework with other sub-block coding schemes, and an exploration of potential application domains beyond the unsourced MAC problem.

In this paper, we restricted our attention to an idealized channel model which assumes perfect symbol-level synchronization and no fading. With appropriate modifications such as in [28], the compressed sensing part can be made to work even in the presence of asynchronism and fading with no CSIR. Hence, in principle, the proposed scheme can be extended to handle more practical scenarios; however, an evaluation of the scheme under such a practical scenario is left for future work.

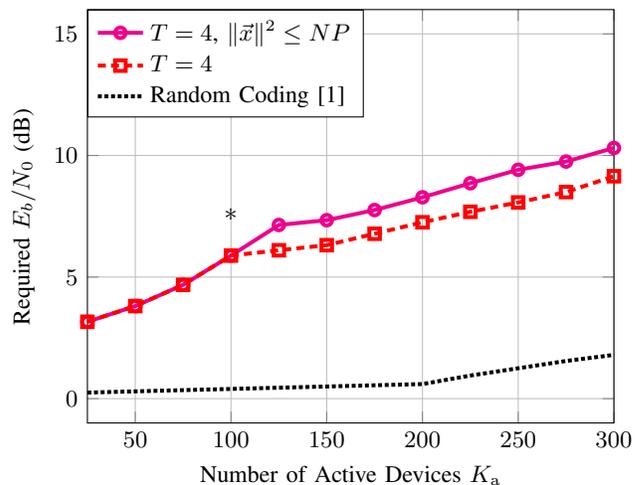


Fig. 9. This plot depicts the minimum  $E_b/N_0$  required to achieve  $P_e \leq 0.05$  as a function of the number of active devices. The curves compare the performance obtained under an average power constraint with that of a system where the power requirement is imposed on every possible transmit signal. The latter requirement is achieved by making sure that the number of times a sub-block codeword is repeated remains constant across messages. We observe that the performance for these two cases remain close.

## REFERENCES

- [1] Y. Polyanskiy, "A perspective on massive random-access," in *Proc. Int. Symp. on Information Theory*, pp. 2523–2527, 2017.
- [2] X. Chen, T.-Y. Chen, and D. Guo, "Capacity of Gaussian many-access channels," *IEEE Trans. Inform. Theory*, vol. 63, no. 6, pp. 3516–3539, 2017.
- [3] B. Rimoldi and R. Urbanke, "A rate-splitting approach to the Gaussian multiple-access channel," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 364–375, 1996.
- [4] L. Ping, L. Liu, K. Wu, and W. K. Leung, "Interleave division multiple-access," *IEEE Trans. on Wireless Comm.*, vol. 5, no. 4, pp. 938–947, 2006.
- [5] O. Ordentlich and Y. Polyanskiy, "Low complexity schemes for the random access Gaussian channel," in *Proc. Int. Symp. on Information Theory*, pp. 2528–2532, 2017.
- [6] J. Zhan, B. Nazer, U. Erez, and M. Gastpar, "Integer-forcing linear receivers," *IEEE Trans. Inform. Theory*, vol. 60, no. 12, pp. 7661–7685, 2014.
- [7] P. Mathys, "A class of codes for a T active users out of N multiple-access communication system," *IEEE Trans. Inform. Theory*, vol. 36, no. 6, pp. 1206–1219, 1990.
- [8] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [9] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*. Birkhäuser Basel, 2013.
- [10] G. Liva, "Graph-based analysis and optimization of contention resolution diversity slotted ALOHA," *IEEE Trans. on Commun.*, vol. 59, no. 2, pp. 477–487, 2011.
- [11] X. Chen and D. Guo, "Many-access channels: The Gaussian case with random user activities," in *Proc. Int. Symp. on Information Theory*, pp. 3127–3131, 2014.
- [12] J. Goseling, M. Gastpar, and J. H. Weber, "Random access with physical-layer network coding," *IEEE Trans. Inform. Theory*, vol. 61, no. 7, pp. 3670–3681, 2015.
- [13] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, no. 5, pp. 533–547, 1981.
- [14] N. C. Wormald, "Differential equations for random processes and random graphs," *Annals of Applied Prob.*, pp. 1217–1235, 1995.
- [15] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: Model and erasure channel properties," *IEEE Trans. Inform. Theory*, vol. 50, no. 11, pp. 2657–2673, 2004.
- [16] T. Richardson and R. Urbanke, *Modern coding theory*. Cambridge University Press, 2008.
- [17] E. Paolini, G. Liva, and M. Chiani, "Coded slotted ALOHA: A graph-based method for uncoordinated multiple access," *IEEE Trans. Inform. Theory*, vol. 61, no. 12, pp. 6815–6832, 2015.
- [18] K. R. Narayanan and H. D. Pfister, "Iterative collision resolution for slotted ALOHA: An optimal uncoordinated transmission policy," in *Proc. Intl. Symp. Turbo Codes and Rel. Topics.*, pp. 136–139, IEEE, 2012.
- [19] E. Paolini, C. Stefanovic, G. Liva, and P. Popovski, "Coded random access: applying codes on graphs to design random access protocols," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 144–150, 2015.
- [20] S. Kudekar, T. Richardson, and R. Urbanke, "Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC," *IEEE Trans. Inform. Theory*, vol. 57, no. 2, pp. 803–834, 2011.
- [21] T. W. Hungerford, "Algebra," 1980.
- [22] Č. Stefanović, D. Vukobratović, J. Goseling, and P. Popovski, "Identifying randomly activated users via sign-compute-resolve on graphs," in *Proc. IEEE International Conf. Commun.*, pp. 589–595, 2016.
- [23] J. Goseling, Č. Stefanović, and P. Popovski, "Sign-compute-resolve for tree splitting random access," *Proc. Int. Symp. on Information Theory*, vol. 64, no. 7, pp. 5261–5276, 2018.
- [24] G. Cocco, C. Ibars, D. Gunduz, and O. del Rio Herrero, "Collision resolution in slotted ALOHA with multi-user physical-layer network coding," in *Proc. Vehicular Tech. Conf.*, pp. 1–4, IEEE, 2011.
- [25] G. Wunder, Č. Stefanović, P. Popovski, and L. Thiele, "Compressive coded random access for massive MTC traffic in 5G systems," in *Proc. Asilomar Conf. Signals, Systems, Comp.*, pp. 13–17, 2015.
- [26] G. Wunder, H. Boche, T. Strohmmer, and P. Jung, "Sparse signal processing concepts for efficient 5G system design," *IEEE Access*, vol. 3, pp. 195–208, 2015.
- [27] F. Fazel, M. Fazel, and M. Stojanovic, "Random access compressed sensing over fading and noisy communication channels," *IEEE Trans. on Wireless Comm.*, vol. 12, no. 5, pp. 2114–2125, 2013.
- [28] V. K. Amalladinne, K. R. Narayanan, J.-F. Chamberland, and D. Guo, "Asynchronous neighbor discovery using coupled compressive sensing," in *Proc. Intl. Conf. Acoustics Speech and Sig. Proc.*, pp. 4569–4573, 2019.

APPENDIX A  
PROOF OF LEMMA 1

This section contains a proof of Lemma 1. We need to show that, for a random random ensemble,

$$\Pr(\mathcal{E}_{\text{corr}}) \leq \exp\left(-\frac{N_p P_p}{2(2 + (2T - 1)P_p)}\right)$$

and, for a binary ensemble,

$$\Pr(\mathcal{E}_{\text{corr}}) \leq \exp\left(-\frac{P_p d_{\max}}{2} \left(1 - T \left(1 - \frac{d_{\min}}{d_{\max}}\right)\right)^2\right).$$

To simplify the exposition, we drop sub-block index  $j$  throughout. We also use the shorthand notation  $\vec{y}_p = \vec{y}[1 : P_p]^T$  and  $\vec{z}_p = \vec{x}[1 : P_p]^T$ . Suppose  $k \in \mathcal{P}$  and  $k' \notin \mathcal{P}$  are fixed. Then, the error event of (17) can be rewritten as

$$\begin{aligned} 0 &\geq \langle \vec{y}_p, \vec{a}_k - \vec{a}_{k'} \rangle \\ &= \langle \vec{z}_p, \vec{a}_k - \vec{a}_{k'} \rangle + N_p P_p - \langle \vec{a}_k, \vec{a}_{k'} \rangle \\ &\quad + \left\langle \sum_{\kappa \in \mathcal{P} \setminus k} \vec{a}_\kappa, \vec{a}_k - \vec{a}_{k'} \right\rangle. \end{aligned} \quad (34)$$

Equivalently, this event can be expressed as

$$\begin{aligned} \langle \vec{z}_p, \vec{a}_{k'} - \vec{a}_k \rangle + \sum_{\kappa \in \mathcal{P} \setminus k} \langle \vec{a}_\kappa, \vec{a}_{k'} - \vec{a}_k \rangle + \langle \vec{a}_k, \vec{a}_{k'} \rangle \\ \geq N_p P_p. \end{aligned} \quad (35)$$

Then, the probability of an error can be computed as

$$\begin{aligned} \Pr(\mathcal{E}_{\text{corr}}) \\ = \Pr\left(\frac{1}{N_p P_p} \left(\langle \vec{z}_p, \vec{a}_{k'} - \vec{a}_k \rangle + \langle \sum_{\kappa \in \mathcal{P} \setminus k} \vec{a}_\kappa, \vec{a}_{k'} - \vec{a}_k \rangle \right. \right. \\ \left. \left. - \langle \sum_{\kappa \in \mathcal{P} \setminus k} \vec{a}_\kappa, \vec{a}_k \rangle\right) \geq 1\right). \end{aligned} \quad (36)$$

Moving forward, we examine the two matrix ensembles separately.

**Random Ensemble:** For Rademacher random ensemble, we note that

$$\mathbb{E} \left[ \langle \vec{z}_p, \vec{a}_{k'} - \vec{a}_k \rangle + \sum_{\kappa \in \mathcal{P} \setminus k} \langle \vec{a}_\kappa, \vec{a}_{k'} - \vec{a}_k \rangle + \langle \vec{a}_k, \vec{a}_{k'} \rangle \right] = 0.$$

Moreover, we have

$$\begin{aligned} \mathbb{E} \left[ \left( \langle \vec{z}_p, \vec{a}_{k'} - \vec{a}_k \rangle + \sum_{\kappa \in \mathcal{P} \setminus k} \langle \vec{a}_\kappa, \vec{a}_{k'} - \vec{a}_k \rangle + \langle \vec{a}_k, \vec{a}_{k'} \rangle \right)^2 \right] \\ = \mathbb{E} \left[ \left( \langle \vec{z}_p, \vec{a}_{k'} - \vec{a}_k \rangle \right)^2 \right] + \mathbb{E} \left[ \left( \sum_{\kappa \in \mathcal{P} \setminus k} \langle \vec{a}_\kappa, \vec{a}_{k'} - \vec{a}_k \rangle \right)^2 \right] \\ + \mathbb{E} \left[ \left( \langle \vec{a}_k, \vec{a}_{k'} \rangle \right)^2 \right] + 2\mathbb{E} \left[ \langle \vec{z}_p, \vec{a}_{k'} - \vec{a}_k \rangle \langle \vec{a}_k, \vec{a}_{k'} \rangle \right] \\ + 2\mathbb{E} \left[ \langle \vec{z}_p, \vec{a}_{k'} - \vec{a}_k \rangle \left( \sum_{\kappa \in \mathcal{P} \setminus k} \langle \vec{a}_\kappa, \vec{a}_{k'} - \vec{a}_k \rangle \right) \right] \\ + 2\mathbb{E} \left[ \left( \sum_{\kappa \in \mathcal{P} \setminus k} \langle \vec{a}_\kappa, \vec{a}_{k'} - \vec{a}_k \rangle \right) \langle \vec{a}_k, \vec{a}_{k'} \rangle \right] \\ = 2N_p P_p + 2(T - 1)N_p P_p^2 + N_p P_p^2 \\ = 2N_p P_p + (2T - 1)N_p P_p^2. \end{aligned}$$

We emphasize that the expectations of the cross-terms are zero. The probability of error can then be bounded using the Chebyshev inequality,

$$\Pr(\mathcal{E}_{\text{corr}}) \leq \frac{(2T+1)N_p P_p^2}{N_p^2 P_p^2} = \frac{2+(2T-1)P_p}{N_p P_p}. \quad (37)$$

For large values of  $N_p$ , the random empirical averages  $\frac{1}{N_p P_p} \langle \sum_{\kappa \in \mathcal{P}} \vec{a}_\kappa, \vec{a}_{k'} \rangle$  and  $\frac{1}{N_p P_p} \langle \sum_{\kappa \in \mathcal{P} \setminus k} \vec{a}_\kappa, \vec{a}_k \rangle$  are independent, and their respective distributions are well approximated by  $\mathcal{N}\left(0, \frac{T}{N_p}\right)$  and  $\mathcal{N}\left(0, \frac{T-1}{N_p}\right)$ . Likewise, the distribution of  $\frac{1}{N_p P_p} \langle \vec{z}_p, \vec{a}_{k'} - \vec{a}_k \rangle$  is well approximated by  $\mathcal{N}\left(0, \frac{2}{N_p P_p}\right)$ . This follows from the law of total variance and a central limit argument. Applying properties of Gaussian random variables, we get

$$\Pr(\mathcal{E}_{\text{corr}}) \leq \exp\left(-\frac{N_p P_p}{2(2+(2T-1)P_p)}\right).$$

*Binary Ensemble:* For this second ensemble, the correlation between any two vectors is bounded by

$$N_p - 2d_{\text{max}} \leq \frac{1}{P_p} \langle \vec{a}_k, \vec{a}_{k'} \rangle \leq N_p - 2d_{\text{min}}.$$

Accordingly, we can write

$$\begin{aligned} & \Pr(\mathcal{E}_{\text{corr}}) \\ &= \Pr\left(\langle \vec{z}_p, \vec{a}_{k'} - \vec{a}_k \rangle + \langle \sum_{\kappa \in \mathcal{P}} \vec{a}_\kappa, \vec{a}_{k'} \rangle \right. \\ & \quad \left. - \langle \sum_{\kappa \in \mathcal{P} \setminus k} \vec{a}_\kappa, \vec{a}_k \rangle \geq N_p P_p\right) \\ &\leq \Pr\left(\langle \vec{z}_p, \vec{a}_{k'} - \vec{a}_k \rangle + P_p T(N_p - 2d_{\text{min}}) \right. \\ & \quad \left. - P_p(T-1)(N_p - 2d_{\text{max}}) \geq N_p P_p\right) \\ &= \Pr(\langle \vec{z}_p, \vec{a}_{k'} - \vec{a}_k \rangle \geq 2P_p(d_{\text{max}} - T(d_{\text{max}} - d_{\text{min}}))). \end{aligned}$$

The variance of  $\langle \vec{z}_p, \vec{a}_{k'} - \vec{a}_k \rangle$  is equal to  $P_p \mathbb{E}[\|\vec{a}_{k'} - \vec{a}_k\|_2^2] \leq 4P_p d_{\text{max}}$ . Again, using properties of the Gaussian distribution, we obtain

$$\Pr(\mathcal{E}_{\text{corr}}) \leq \exp\left(-\frac{P_p(d_{\text{max}} - T(d_{\text{max}} - d_{\text{min}}))^2}{2d_{\text{max}}}\right).$$