APPLICATIONS OF CODING THEORY TO MASSIVE MULTIPLE ACCESS AND BIG DATA PROBLEMS

A Dissertation

by

AVINASH VEM

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Krishna R. Narayanan
Committee Members,	Arun R. Srinivasa
	Jean-Francois Chamberland
	Alex Sprintson
Head of Department,	Miroslav M. Begovic

December 2017

Major Subject: Electrical Engineering

ABSTRACT

The broad theme of this dissertation is design of schemes that admit iterative algorithms with low computational complexity to some new problems arising in massive multiple access and big data. Although bipartite Tanner graphs and low-complexity iterative algorithms such as peeling and message passing decoders are very popular in the channel coding literature they are not as widely used in the respective areas of study and this dissertation serves as an important step in that direction to bridge that gap. The contributions of this dissertation can be categorized into the following three parts.

In the first part of this dissertation, a timely and interesting multiple access problem for a massive number of uncoordinated devices is considered wherein the base station is interested only in recovering the list of messages without regard to the identity of the respective sources. A coding scheme with polynomial encoding and decoding complexities is proposed for this problem, the two main features of which are (i) design of a *close-to-optimal* coding scheme for the T-user Gaussian multiple access channel and (ii) successive interference cancellation decoder. The proposed coding scheme not only improves on the performance of the previously best known coding scheme by $\approx$ 13 dB but is only $\approx$ 6 dB away from the random Gaussian coding information rate.

In the second part construction-D lattices are constructed where the underlying linear codes are nested binary spatially-coupled low-density parity-check codes (SC-LDPC) codes with uniform left and right degrees. It is shown that the proposed lattices achieve the Poltyrev limit under multistage belief propagation decoding. Leveraging this result lattice codes constructed from these lattices are applied to the

three user symmetric interference channel. For channel gains within 0.39 dB from the very strong interference regime, the proposed lattice coding scheme with the iterative belief propagation decoder, for target error rates of $\approx 10^{-5}$, is only 2.6 dB away the Shannon limit.

The third part focuses on support recovery in compressed sensing and the non-adaptive group testing (GT) problems. Prior to this work, sensing schemes based on left-regular sparse bipartite graphs and iterative recovery algorithms based on peeling decoder were proposed for the above problems. These schemes require $\mathcal{O}(K \log N)$ and $\Omega(K \log K \log N)$ measurements respectively to recover the sparse signal with high probability ($w.h.p$), where N, K denote the dimension and sparsity of the signal respectively ($K \ll N$). Also the number of measurements required to recover atleast $(1-\epsilon)$ fraction of defective items w.h.p (approximate GT) is shown to be $c_\epsilon K \log N$. In this dissertation, instead of the left-regular bipartite graphs, left-and-right regular bipartite graph based sensing schemes are analyzed. It is shown that this design strategy enables to achieve superior and sharper results. For the support recovery problem, the number of measurements is reduced to the optimal lower bound of $\Omega\left(K \log \frac{N}{K}\right)$. Similarly for the approximate GT, proposed scheme only requires $c_\epsilon K \log \frac{N}{K}$ measurements. For the probabilistic GT, proposed scheme requires $\Omega(K \log K \log \frac{N}{K})$ measurements which is only $\log K$ factor away from the best known lower bound of $\Omega(K \log \frac{N}{K})$. Apart from the asymptotic regime, the proposed schemes also demonstrate significant improvement in the required number of measurements for finite values of K, N.

To my loving family,

and in memory of my uncle (1964-2014)

ACKNOWLEDGMENTS

It was a long, rewarding and fulfilling journey to my Ph. D. degree. This would not have been possible if not for the support and encouragement of a lot of people starting from my teachers and friends in high school to my professors, friends and colleagues at Texas A&M university. My heartfelt gratitude and thanks go out to each and everyone who helped me and stood by me.

First and foremost, I would like to thank my advisor Dr. Krishna Narayanan for his guidance throughout the past six years of my graduate studies. He was always available, literally a knock-on-the-door away, with plenty of time for me. I will be forever indebted to his teachings, support and the valuable directions he provided in times of struggle. He taught me coding and information theory courses in my first two semesters at Texas A& M University and his passionate teaching arose the curiosity in me and laid the foundations for my subsequent forays into these rich and beautiful areas of research. I am truly fortunate to have spent the formative years of my academic and professional career under the tutelage of such a wonderful academic and a warm person. I hope to carry his legacy forward.

I would like to thank Professors Jean-Francois Chamberland, Arun Srinivasa, and Alex Sprintson for their time in serving on my thesis committee. I am thankful for their insights.

I am thankful to a lot of friends and colleagues whose company and camaraderie made this journey all the more enjoyable. I want to reminisce and thank my roommates Nilu, Rhushabh 'Sir' for all the late night discussions, Shweta, Suraj, Sneha, Gupta, Kripa *et al.*, from my second-family 'Nagle' group for the innumerable weekends spent in laughter and banter, Manjusha, Kartheek and Siddhitha for their

affection. I am also thankful to my lab-mates Arvind Yedla, for the crazy parties and fun times spent in and 'around' Eugenes, Santhosh Vanaparthy, for the hours long discussions we had on a wide range of topics varying from madness of mathematicians like Paul Dirac to the importance/stupidity of sports etc., Rahul, Emmadi, Jerry Huang, Engin, Yung-Yih Jian, Fatemeh, Nagaraj, Paul, Arman, and several others.

Above all, I am forever grateful to Dakshna for all the wonderful times we spent in College Station, for being with me in the ups and downs of this journey, for providing the care, comfort, motivation, encouragement, guidance and support when I needed them. At a moment of looking back, I want to remember my uncle Narasimha Reddy for his assistance and encouragement in my pursuing higher studies. Although no longer with us, he continues to inspire me by the hard work and love he showed for his family.

Finally, I express my deepest gratitude to my parents, Ranga Reddy and Andalamma, and my little sister Anu for their unconditional love and support and for the immense faith they have always shown in me. I cannot thank enough, my mother for all the sacrifices she made to ensure our happiness, and my father for engaging me in all my inane questions in my childhood, his insistence that I learn the mathematics and the sciences in the *right* way and for prioritizing the higher education of me and my sister above all else. I will be forever indebted to them.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Professor Krishna Narayanan (advisor) and Professors Jean-Francois Chamberland and Alex Sprintson of the Department of Electrical and Computer Engineering, Texas A&M University and Professor Arun Srinivasa of the Department of Mechanical Engineering, Texas A&M University. The work presented in Chapter II is carried out in collaboration with Professor Jun Cheng of the Department of Intelligent Information Engineering and Science, Doshisha University while he was visiting Texas A&M University. The work presented in Chapter IV is carried out in collaboration with Professor Henry Pfister of the Department of Electrical and Computer Engineering, Duke University during his time as Professor in the Department of Electrical and Computer Engineering, Texas A&M University.

All other work conducted for the dissertation was completed by the student independently.

Funding Sources

This work was supported in part by the National Science Foundation (NSF) under Grants No. CCF-13202924, CCF-1302616, EARS-1547447 and CCF-1619085.

NOMENCLATURE

SNR	Signal-to-noise ratio
SIC	Successive interference cancellation
MAC	Mulitple-access
GMAC	Gaussian multiple-access
SC-LDPC	Spatially-coupled low-density parity check
CS	Compressed sensing
AWGN	Additive white Gaussian noise
BP	Belief propagation
DE	Density evolution

-

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

TABLE Page

I. INTRODUCTION

We are entering an era of *massive* by every metric of interest whether it be the total number of internet users, the total number of networked devices or the amount of data that needs to be stored and accessed. By 2020 the total number of connected smart devices in the world (excluding smart phones, tablets and computers) that are embedded with electronics and sensors is estimated to reach 20.8 billion according to Gartner analytics or 28.1 billion according to International Data Corporation (IDC). Similarly in regards to data storage and traffic Cisco estimates that cloud traffic could rise to 14.1 zettabytes (ZB) by 2020 from 3.9 ZB in 2015. Note that 1 ZB=10^{21} bytes=1 trillion gigabytes. According to IDC the total amount of digital data created worldwide could rise to 44 ZB by 2020. The Internet of Things (IoT) and the associated big data are a big part of this growth. By any standard this is a massive number of devices and an enormous amount of data and this provides for exciting opportunities in various engineering and scientific domains like advancing health care, resource utilization patterns, understanding the demands of certain demographics etc., via data-mining. However there are two significant challenges that need to be addressed before any such advances are feasible:

- design of efficient communication protocols for a large number of smart devices
- data mining the available massive data sets in an algorithmically efficient manner.

In this thesis we attempt to tackle some problems that fall under these two issues and provide practical, low-complexity solutions for such problems. In the following section we summarize the contributions of this dissertation.

I.A Organization

I.A.1 Background

The overarching theme of this dissertation is leveraging the sparse bipartite Tanner graph structure and the associated low-complexity iterative peeling decoding algorithms to construct design schemes for the multiple access communication and big data problems outlined above.

I.A.2 Massive multiple access

In Chapters II and III, we consider the massive multiple access problem. The imminent advent of IoT gives rise to a framework consisting of a large number of sensor devices that have brief but sporadic messages to communicate. This poses a vastly different set of challenges for radio resource management in wireless infrastructures. Currently deployed scheduling policies and wireless protocols which are suitable for fairly small number of sustained connections are ill-equipped to deal with such IoT traffic since they rely on gathering information about channel quality and queue length for every active user and hence pose a significant overhead to the system. This paradigm is unsustainable in environments with myriad devices, each sending a brief message. This points to an urgent need for design of practical uncoordinated schemes for the massive multiple access setup.

In the uncoordinated multiple access setup, each device in the system wants to transmit a message of certain length to the access point in an uncoordinated fashion. The total available time for communication is divided into slots of constant length where the users are assumed to know the structure of time slots. The access point is interested in recovering the messages transmitted by each user. In 1970 Abramson in his pioneering work [6] proposed a random access scheme, known as ALOHA, that achieves a throughput of $1/e \approx 0.37$. In the slotted version of the ALOHA

scheme each user repeats the intended message in a certain number of slots, slots being chosen randomly and independently of other users in an uncoordinated fashion. All the slots in which there is no collision i.e., there is only one user transmitting, the message can be decoded successfully and the slots with collisions are simply discarded. This remained the state-of-the-art until a decade ago. In 2007 [7] Cassini *et al* showed that higher throughput can be achieved by not discarding the slots where the transmissions of distinct users *collide* but by using these slots to decode the colliding users via iterative successive interference cancellation (SIC) process.

In 2011, Liva demonstrated a close connection between the analysis of such random access schemes under the SIC decoding process and the design of low density generator matrix codes [8]. Strengthening this connection, in Chapter III we introduce an analytical framework for analyzing the evolution of the iterative SIC process as a function of the random access strategy employed by each user in the system. In 2012, Narayanan and Pfister showed that by choosing the repetition parameter randomly according to a Soliton distribution and using SIC decoder the optimal throughput of one can be achieved asymptotically[9]. However, this paradigm of choosing according to Soliton distribution is known to perform poorly when the number of active devices is not very large. We took the first step in addressing this issue. In Chapter III, given a probability distribution with finite maximum degree (not necessarily Soliton), we provide analytic expressions to compute the probability of error for the SIC decoder in the random uncoordinated access problem. The analytic evaluation of the error performance offers a possible solution path to designing optimal random access strategies for this practical setup.

The unsourced formulation of the massive multiple access corresponds to the scenario where an access point only wishes to recover the collection of sent messages, and not the identity of the respective sources. Although the sourced formulation of

the uncoordinated multiple access described earlier has been around for nearly four decades, Polyanskiy in 2017 for the first time considered the unsourced formulation of the multiple access [2]. Polyanskiy and Ordentlich [10] proposed a coding scheme for the unsourced multiple access based on concatenated codes. The authors show that this scheme outperforms all the other existing schemes available for the uncoordinated multiple access. In Chapter II we propose a coding scheme for the unsourced MAC in which the transmitted codeword is purely a function of the message being transmitted thus exploiting the unsourced nature of the problem. The main differentiating ingredient in our scheme when compared to [10] is that we use successive interference cancellation decoding process. We show that our proposed scheme not only improves substantially on the performance in [10] but is also only $\approx$ 6dB away from the achievable limit based on random Gaussian coding and joint typical decoder which has exponential complexity [2].

I.A.3 Interference channel

While the massive multiple access framework is important in the context of IoT, many-to-many communication setups with a small number of users such as Gaussian interference channel are still relevant. Finding the capacity of the Gaussian interference channel has been a long standing open problem in information theory[11]. The capacity is derived under certain conditions such as (i) two-user interference channel with *very strong* interference [12, 13], (ii) characterization of capacity region to within one bit per channel use [14] (iii) *approximate* characterization of many-to-one and one-to-many interference channels etc. Since we do not yet know the characterization of the full capacity region few attempts were made at designing practical coding schemes for the Gaussian interference channel. In [15] it was shown that lattice coding achieves the capacity of the two-user symmetric interference channel

under very strong interference. However no practical lattice coding schemes were provided. In Chapter IV we attempt to bridge this gap by constructing a new class of lattices using construction-D where the underlying linear codes are nested binary spatially-coupled low-density parity-check codes (SC-LDPC) codes from the uniform left and right degree ensembles. By leveraging results on the optimality of spatially-coupled codes for binary input memoryless symmetric channels and Forney *et al.*'s earlier results on the optimality of construction-D, we show that the proposed lattices achieve the Poltyrev limit under low-complexity iterative multistage belief propagation decoding. We then show that the lattice codes derived from the proposed lattices via hyper cube shaping perform upto a shaping loss of 1.53dB for the three user symmetric interference channel.

In Chapters V & VI we focus on the sparse signal estimation problems.

I.A.4 Compressed sensing

Compressed sensing is a signal processing technique for efficiently acquiring linear measurements, traditionally referred to as *sensing*, of a sparse signal and reconstructing the signal from the acquired measurements. In Chapter V, we focus on the support recovery problem in compressed sensing wherein the objective is to recover the set of signal dimensions with non-zero power and not necessarily the whole signal. In 2015 Li, Pawar and Ramchandran proposed two schemes to recover the support of a K-sparse N-dimensional signal from noisy linear measurements [4, 16]. Both the schemes employ left-regular sparse bipartite graph code based matrices for sensing the signal and a peeling based reconstruction algorithm. Both the schemes require $O(K \log N)$ measurements and the first scheme requires $O(N \log N)$ total computations whereas the second scheme requires $O(K \log N)$ total computations (sub-linear computational complexity when K is sub-linear in N). We show that

by replacing the left-regular ensemble with left-and-right regular ensemble, we can reduce the number of measurements required of these schemes to the optimal order of $\Theta\left(K \log \frac{N}{K}\right)$ with optimal decoding complexities of $O(K \log \frac{N}{K})$ and $O(N \log \frac{N}{K})$ respectively.

I.A.5 Group testing

The group testing problem was first introduced to the fields of applied mathematics and statistics by Dorfman [17] during World War II for testing the soldiers for syphilis without having to test each soldier individually. The aim of the problem is to detect K defective items out of a large population of N total items where grouping multiple items together for a single test is possible. The output of the test is *negative* if all the grouped items are non-defective or else the output is *positive*. In Chapter VI, we focus on the non-adaptive version of group testing where the testing scheme is pre-determined and is independent of the test results. We propose a testing scheme based on left-and-right regular sparse bipartite graphs that admit a simple iterative recovery scheme and show that for any arbitrarily small $\epsilon > 0$ our scheme requires only $m = c_\epsilon K \log \frac{c_1 N}{K}$ tests to recover $(1 - \epsilon)$ fraction of the defective items with high probability (w.h.p) i.e., with probability approaching 1 asymptotically in N and K, where the value of constants c_ϵ and ℓ are a function of the desired error floor ϵ and constant $c_1 = \frac{\ell}{c_\epsilon}$ (observed to be approximately equal to 1 for various values of ϵ). More importantly the iterative decoding algorithm has a sub-linear computational complexity of $O(K \log \frac{N}{K})$ which is known to be optimal. Also for $m = c_2 K \log K \log \frac{N}{K}$ tests our scheme recovers the *whole* set of defective items w.h.p. These results are valid for both noiseless and noisy versions of the problem as long as the number of defective items scale sub-linearly with the total number of items, i.e., $K = o(N)$. The simulation results validate the theoretical results by

showing a substantial improvement in the number of tests required when compared to the testing scheme based on the left regular sparse graphs.

II. MASSIVE MULTIPLE ACCESS*

In [2], Polyanskiy introduced an interesting and timely multiple access problem; throughout, we refer to this new formulation as the unsourced multiple access channel model (MAC). In this setting, a very large number, K_{tot}, of users in a wireless network operate in an uncoordinated fashion. Out of the K_{tot} users, a subset of K_{a} users are active at any time; and each of them wishes to communicate a B-bit message to a central base station. The base station is interested only in recovering the list of messages without regard to the identity of the user who transmitted a particular message. In addition to this, the interest is typically in the case when B is small.

The unsourced, uncoordinated nature of the problem and the small block lengths represent a substantial departure from the traditional multiple access channel and, consequently, has important implications both on the fundamental limits as well as the design of pragmatic low-complexity coding schemes. Due to small block lengths, information rates do not provide reasonable benchmarks and finite block length bounds are more meaningful. In [2], Polyanskiy provides bounds on the performance of finite-length codes for this channel model. The design of coding schemes is also very challenging for this setting. Almost all well-known low-complexity coding solutions for the traditional MAC channel such as code-division multiple access, rate-splitting [18], and interleave-division multiple access [19], implicitly assume some form of coordination between the users and that some parameters of the coding scheme such as the spreading sequence, code rates, time sharing parameters, Tanner graph of the code, etc., are user dependent. When the message length is small, establishing such

coordination becomes inefficient; this renders well-known coding solutions tailored to the traditional MAC inadequate for the unsourced MAC. Ordentlich and Polyanskiy describe the first low-complexity coding paradigm for the unsourced MAC [10]. In their scheme, a transmission period is partitioned into smaller sub-blocks and users randomly pick one sub-block to transmit in. The encoding structure employed by each user is a concatenated code where the inner code is designed to recover the modulo-p sum of codewords transmitted by users and the outer code is designed to decode multiple users given the modulo-p sum of their codewords. Succinctly, the inner code operates in the spirit of integer-forcing [20], whereas the outer code is an optimal code for the T-user modulo-p multiple access channel [21].

While Ordentlich and Polyanskiy have contributed an important first step in finding practical schemes for the unsourced MAC, there remains a substantial gap between the performance of their proposed scheme and the capacity limit derived in [2]. Indeed, they point to this gap and discuss possibilities for improving its performance. In [10, 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. However, in the unsourced MAC, each user is forced to use the same codebook and they remark that "the task of designing low complexity capacity approaching same-codebook schemes for the real binary adder seems quite challenging." Another important limitation that is not discussed in [10] is that their scheme does not admit iterative cancellation and, hence, successive interference cancellation is not considered. Therefore, when more than T-users transmit in a slot, this slot is not utilized in the decoding process. As a result, their scheme uses a large number of slots in order to ensure that every user is received in a time slot that contains at most T-users, resulting in poor spectral efficiency.

The main contribution of this chapter is to propose, analyze and optimize a new coding architecture that overcomes these drawbacks and substantially improves performance when compared to the state-of-the-art. Key features of our scheme are summarized as follows.

- **User symmetry:** Active users employ the same coding scheme, with transmitted signals determined solely by the message to be transmitted and is independent of the identity of the user. To be precise, no parameter of the encoding scheme such as the interleaver and spreading sequence are unique to a transmitter.

- **Binary-input, real-adder channel:** The proposed coding scheme is tailored to the binary-input real-adder channel. The information message is split into two parts. The first portion picks an 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 compressed sensing scheme. The second part is decoded using a message passing decoder that jointly recovers up to T messages within a slot.

- **Successive interference cancellation:** Active users repeat their codewords in several slots. The repetition patterns are selected based on message bits. This scheme facilitates interference cancellation within the slotted structure, and therefore renders obsolete the over-provisioning of slots to avoid undue collisions with more than T users.

While [10] also incorporates the user symmetry aspect described above, our scheme differs from theirs in the other features highlighted above.

Notation	Parameter represented
K_{tot}	Total number of users in the system
K_{a}	Number of active users
$\tilde{N}$	Number of channel uses per frame
ϵ	Maximum decoding probability of error, per active user
V	Number of slots each frame is divided into
N	Number of channel uses per slot i.e. $N = \tilde{N}/V$
B	Number of message bits each active user wants to transmit
$N_{\text{p}}, N_{\text{c}}$	Channel uses allocated for preamble and channel coding respectively. Note that $N_{\text{p}} + N_{\text{c}} = N$
$B_{\text{p}}, B_{\text{c}}$	Message bits transmitted by the preamble and channel coding components respectively. Note that $B_{\text{p}} + B_{\text{c}} = B$

Table II.1: Important parameters encountered in this chapter along with the notation used are listed above.

II.A System model

The observed signal vector at the receiver corresponding to the $\tilde{N}$ channel uses can be written as

$$\vec{\text{y}} = \sum_{i=1}^{K_{\text{tot}}} s_i \vec{\text{x}}_i + \vec{\text{z}}, \tag{II.1}$$

where $\vec{\text{x}}_i$ is a signal of dimension $\tilde{N}$ transmitted by the user i, and the additive noise is characterized by $\vec{z} \sim \mathcal{N}(0, \mathbf{I}_{\tilde{N}})$. For convenience, we use boolean indicators indexed by i, where $s_i = 1$ if user i is active and $s_i = 0$ otherwise. We impose an average power constraint on the transmitted vectors when averaged over all possible message indices, i.e., $\frac{1}{M}\sum_w ||\vec{\text{x}}(w)||^2 \leq \tilde{N}P$. The receiver produces a list of messages $\mathcal{L}(\vec{\text{y}}) = \{\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_{K_{\text{a}}}\}$. As in [10], the probability of decoding error, per active user, is defined as

$$P_e = \max_{|(s_1,\ldots,s_{K_{\text{tot}}})|=K_{\text{a}}} \frac{1}{K_{\text{a}}} \sum_{i=1}^{K_{\text{tot}}} s_i \Pr\left(w_i \notin \mathcal{L}(\vec{\text{y}})\right) \tag{II.2}$$

where $|\cdot|$ denotes the Hamming weight. The objective of the problem is to design a coding scheme with polynomial encoding and decoding complexities such that $P_e \leq \epsilon$ for a given per user target error rate ϵ.

II.B Description of the proposed scheme

The overall schematic of the proposed scheme is shown in Fig. II.1. In our proposed scheme, the $\tilde{N}$ channel uses which are available for communication are split into V sub-blocks (also referred to as slots throughout the chapter), each of length $N = \tilde{N}/V$ channel uses. The encoding operation at the i-th user takes place in two steps.

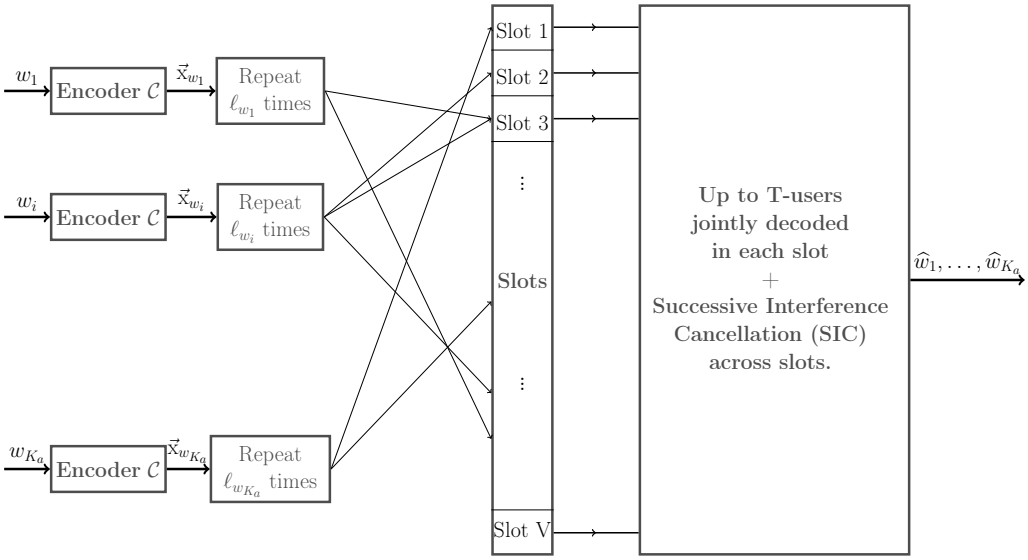

Figure II.1: Schematic of the proposed scheme

II.B.1 Transmission policy across sub-blocks - message based repetition

For the code word to be transmitted in a sub-block each user uses an identical code book (not-necessarily linear) $\mathcal{C}$ of rate $\frac{B}{N}$ and length N. Given the message index

to be transmitted is w_i the user encodes it into a codeword $\vec{c}_{w_i} \in \mathcal{C}$ and modulates $\vec{c}_{w_i}$ into $\vec{x}_{w_i}$. In the following discussion, we will refer to $\vec{c}_{w_i}$ as the transmitted codeword and the reader should assume that the codeword is modulated appropriately and transmitted. Each user also chooses a repetition parameter $\ell_{w_i} = g(w_i)$ using a function $g : [1 : M] \to [1 : V]$ and repeats their codeword $\vec{c}_{w_i}$, ℓ_{w_i} times by choosing ℓ_{w_i} sub blocks from $[1 : V]$ based on the message w_i and transmits during these sub blocks. It is important to note that ℓ_{w_i} as well as the slots where the codeword is repeated are deterministic functions of the message index and do not depend on the identity of the user. As shown in Fig. II.1, a Tanner graph G can be used to visualize the repetition of the codewords where the left nodes correspond to users and the right nodes corresponds to sub-blocks. The degree of the left nodes is determined by ℓ_{w_i} and choosing w_i uniformly at random induces a distribution on ℓ_{w_i} through the function g. Let the left degree distribution (d.d) from node perspective be $L(x) = \sum_{i=1}^{l_{\max}} L_i x^i$, where L_i denotes the fraction of user (left) nodes that are connected to i slot(right) nodes. Similarly let the left d.d from edge perspective be denoted by $\lambda(x) = \sum_{i=1}^{l_{\max}} \lambda_i x^{i-1}$, where λ_i denotes the fraction of edges in G that are connected to left nodes connected to $i-1$ other edges. The two distributions $L(x)$ and $\lambda(x)$ are related as $L(x) = \frac{L'(x)}{L'(1)}$. We choose the mapping g such that a desired left d.d. $L(x)$ (or equivalently $\lambda(x)$) is obtained.

During the j-th sub-block, let $\mathcal{N}_j$ denote the set of users who transmit. During the j-th sub-block, the i-th user transmits symbols of positive power if $i \in \mathcal{N}_j$. Otherwise, the i-th user remains silent. The received signal during the j-th sub-block is given by

$$\vec{\mathrm{y}}_j = \sum_{i \in \mathcal{N}_j} \vec{\mathrm{x}}_{w_i} + \vec{\mathrm{z}}_j. \tag{II.3}$$

II.B.2 Transmission policy within a sub-block - same code book scheme for the T-user multiple access

There are two components to the code $\mathcal{C}$ used in the proposed transmission scheme within each sub-block: a good sensing matrix for a T-sparse robust compressed sensing (CS) problem and a good channel code for the T-user binary-input real-adder channel that is decodable with low computational complexity. The B bits to be transmitted are split into two groups of size B_{p} and $B_{\mathrm{c}} = B - B_{\mathrm{p}}$ bits, respectively. For convenience, we define $M_{\mathrm{p}} := 2^{B_{\mathrm{p}}}$ and $M_{\mathrm{c}} := 2^{B_{\mathrm{c}}}$. The main idea is to use a linear code $\mathcal{C}_{\mathrm{c}}$ good for multiple access channel coding to encode B_{c} message bits which we refer to as channel coding message bits. The remaining B_{p} bits, which we refer to as preamble message bits, are used to pick a permutation of the codeword belonging to the channel code $\mathcal{C}_{\mathrm{c}}$ encoded using the B_{c} channel coding message bits. Typically, we want $B_{\mathrm{p}} \ll B_{\mathrm{c}}$.

For the channel coding part of the code book $\mathcal{C}$ 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 $\mathcal{C}_{\mathrm{c}}$ of rate $\frac{B_{\mathrm{c}}}{N_{\mathrm{c}}}$ and length N_{c}. As an example, we will consider the case when $\mathcal{C}_{\mathrm{c}}$ is chosen uniformly at random from the $(\mathsf{l}, \mathsf{r}, \mathsf{w}, \mathsf{N}_{\mathrm{c}})$ SCLDPC ensemble [1]. Let the modulated codewords of $\mathcal{C}_{\mathrm{c}}$ be denoted by $\{\vec{\mathrm{c}}_1, \vec{\mathrm{c}}_2, \ldots, \vec{\mathrm{c}}_{M_{\mathrm{c}}}\}$, where $\vec{\mathrm{c}}_w = [c_w(1), c_w(2), \ldots, c_w(N_{\mathrm{c}})]$, $c_w(i) \in \{\pm\sqrt{P_{\mathrm{c}}}\}$ $\forall i$ satisfying the power constraint

$$||\vec{\mathrm{c}}_w||_2^2 = N_{\mathrm{c}} P_{\mathrm{c}} \tag{II.4}$$

denotes the modulated SCLDPC codeword corresponding to message index w.

For the second part of the encoder let $\mathbf{A} \in \{-\sqrt{P_{\mathrm{p}}}, +\sqrt{P_{\mathrm{p}}}\}^{N_{\mathrm{p}} \times M_{\mathrm{p}}}$ denote a sensing matrix that can recover the sum of any T columns of $\mathbf{A}$ with low error

probability. Let $f : [1 : M_{\mathrm{p}}] \rightarrow [1 : N_{\mathrm{c}}!]$ denote a hash function which maps B_{p} preamble message bits into an integer $\tau_w = f(w)$ such that τ_w is uniformly distributed over $[1 : N_{\mathrm{c}}!]$ denoting all possible permutations of length N_{c}. Note that here the integer τ_w chooses the permutation $\pi_{\tau_w} \in S_{N_{\mathrm{c}}}$ of the encoded codeword from $\mathcal{C}_{\mathrm{c}}$ before transmission where $S_{N_{\mathrm{c}}}$ is the symmetric group.

The description of the overall encoder for code book $\mathcal{C}$ combining the above two components can be described as following. Let $w = (w^{\mathrm{p}}, w^{\mathrm{c}})$ be the message index to be encoded, where the indices w^{p} and w^{c} correspond to the preamble and coding message indices respectively. We first encode the message index w^{c} to the codeword $\vec{\mathrm{c}}_{w^{\mathrm{c}}} \in \mathcal{C}_c$ followed by permuting it according to permutation $\pi_{\tau_{w^{\mathrm{p}}}} = [\pi^1_{\tau_{w^{\mathrm{p}}}}, \pi^2_{\tau_{w^{\mathrm{p}}}}, \ldots, \pi^{N_{\mathrm{c}}}_{\tau_{w^{\mathrm{p}}}}]$. The final code word $\vec{\mathrm{c}}_w$ is then obtained by inserting the w^{p}th column from the compressed sensing matrix $\mathbf{A}$ at the beginning of the permuted codeword i.e.,

$$\begin{aligned} \vec{\mathrm{c}}_w &= [\vec{\mathrm{a}}_{w^{\mathrm{p}}}, \pi_{\tau_{w^{\mathrm{p}}}}(\vec{\mathrm{c}}_{w^{\mathrm{c}}})] \qquad\qquad \text{where } \vec{\mathrm{a}}_{w^{\mathrm{p}}} \in \mathbf{A}, \vec{\mathrm{c}}_{w^{\mathrm{c}}} \in \mathcal{C}_{\mathrm{c}} \\ &= [\vec{\mathrm{a}}_{w^{\mathrm{p}}}, c_{w^{\mathrm{c}}}(\pi^1_{\tau_{w^{\mathrm{p}}}}), c_{w^{\mathrm{c}}}(\pi^2_{\tau_{w^{\mathrm{p}}}}), \ldots, c_{w^{\mathrm{c}}}(\pi^{N_{\mathrm{c}}}_{\tau_{w^{\mathrm{p}}}})]. \end{aligned} \tag{II.5}$$

The overall encoding process is summarized in Fig. II.2.

The main idea here is that permuting the codeword $\vec{\mathrm{c}}_{w^{\mathrm{c}}}$ decorrelates the multiple access interference from users even though they use identical linear codes and results in a performance that is similar to that obtained by using different codes of identical rates for the different users. This is similar to interleave-division multiple access scheme that was originally proposed in [19]. The overall code is non-linear because of the random permutations for different codewords and $\vec{\mathrm{a}}_{w^{\mathrm{p}}}$ being appended at the beginning. However, if $\vec{\mathrm{a}}_{w^{\mathrm{p}}}$ is identified (and consequently also w^{p}) and removed at the receiver, then the permutations can be determined and decoding the users can

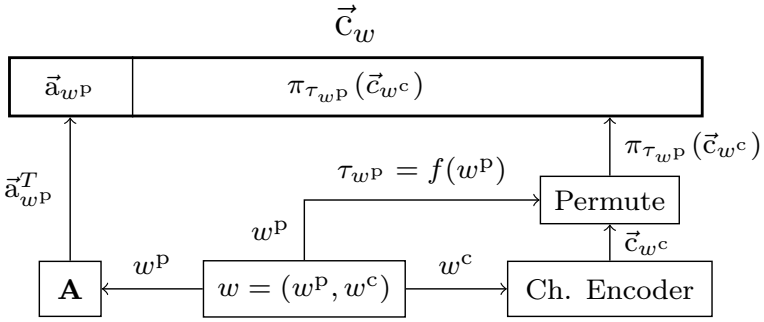

Figure II.2: Schematic depicting the overall encoding scheme in a sub-block given the message index $w = (w^{\mathrm{p}}, w^{\mathrm{c}})$. The final code word transmitted in a sub-block is given by $\vec{c}_w = [\vec{a}_{w^{\mathrm{p}}}, \pi_{\tau_{w^{\mathrm{p}}}}(\vec{c}_{w^{\mathrm{c}}})]$.

be accomplished using a belief propagation decoder that works on the joint graph of the two users.

The overall decoder has two components - a decoder for the T-user Gaussian multiple access(GMAC) channel that works within a sub-block and a serial interference canceler that works across sub-blocks. Note that T is a design parameter of choice. The code book $\mathcal{C}$ within a sub-block is designed such that if T or less users transmit simultaneously within a sub-block the set of the respective transmitted code words can be decoded with low probability of error.

In the following sub-sections we first describe the decoding process within each sub-block followed by the SIC decoding process that works across sub-blocks.

II.B.3 Decoding process within a sub-block

The decoder first estimates the number of users transmitted in a sub-block. Let $R_j = |\mathcal{N}_j|$ denote the number of users that have transmitted during the j-th sub-block. Given $\vec{y}_j$ is the received vector during sub-block j, a simple estimate for R_j

based on energy of the received vector is given by

$$\hat{R}_j = \left[\frac{||\vec{\mathrm{y}}_j||^2 - N\sigma^2}{N_{\mathrm{c}}P_{\mathrm{c}} + N_{\mathrm{p}}P_{\mathrm{p}}} \right]$$

where $[\cdot]$ denotes the nearest integer function and the noise variance $\sigma^2 = 1$ through out this paper. Although the simple energy based estimate is adequate for the scope of this paper, more sophisticated estimates based on GMAC decoding can be obtained, if necessary.

The received signal $\vec{\mathrm{y}}_j$ in sub-block j is

$$\begin{aligned} \vec{\mathrm{y}}_j &= \sum_{i \in \mathcal{N}_j} \vec{\mathrm{x}}_{w_i} + \vec{\mathrm{z}}_j \\ &= \sum_{i \in \mathcal{N}_j} [\vec{\mathrm{a}}_{w_i^{\mathrm{p}}} \quad \pi_{\tau_{w_i^{\mathrm{p}}}}(\vec{\mathrm{c}}_{w_i^{\mathrm{c}}})] + \vec{\mathrm{z}}_j. \end{aligned}$$

As discussed earlier, since the codebook $\mathcal{C}$ employed within the sub-block is designed for T-user GMAC channel the decoder aims to recover the set of messages $\{w_i = (w_i^{\mathrm{p}}, w_i^{\mathrm{c}}), i \in \mathcal{N}_j\}$ and equivalently the set of transmitted codewords $\{\vec{\mathrm{x}}_{w_i}, i \in \mathcal{N}_j\}$ if $|\mathcal{N}|_j \leq T$. There are three components to this decoder: (i) the first component, referred to as compressed sensing (CS) decoder, decodes the set of preamble message indices, (ii) the second component error energy test performs an energy test on the residual error after the compressed sensing decoder to determine whether the output of the compressed sensing decoder in the sub-block is accurate and (iii) the third component, referred to as channel coding decoder, given the set of preamble message indices from the CS decoder as input, decodes the set of channel coding message indices.

Compressed sensing (CS) decoder

The input to the compressed sensing decoder is the preamble component of the received signal given by

$$\vec{y_j}^{\mathrm{p}} := \vec{y}_j[1:N_{\mathrm{p}}] = \sum_{i \in \mathcal{N}_j} \vec{\mathrm{a}}_{w_i^{\mathrm{p}}} + \vec{\mathrm{z}}_j[1:N_{\mathrm{p}}] \tag{II.6}$$

$$= \mathbf{A}\vec{b}_j + \vec{\mathrm{z}}_j^{\mathrm{p}} \tag{II.7}$$

where $\mathbf{A}$ is the sensing matrix and $\vec{b}_j \in \{0,1\}^{M_{\mathrm{p}}}$ is a $|R_j|$-sparse vector that indicates the set of transmitted messages during sub-block j. Our proposed decoder to recover $\vec{b}_j$ from $\vec{y_j}^{\mathrm{p}}$ exploits the sparsity of $\vec{b}_j$ as well as the fact that the non-zero entries of $\vec{b}_j$ are all equal to one. The latter aspect makes the design of the decoder different from many standard compressed sensing reconstruction algorithms.

We consider two options for the choice of compressed sensing decoder. The first option is correlation decoder based on the simple idea that the correlation of the received vector with any of the R_j participating sensing vectors would be high and would be low for the rest.

- *Correlation decoder*: We correlate the preamble part of the received vector with all the columns of the sensing matrix and output the list of $\hat{R}_j$ column indices that have the maximum correlation value:

$$\widehat{\mathcal{W}}_j^{\mathrm{p}} = \arg\max_i \langle \vec{\mathrm{y}}_j^{\mathrm{p}} \vec{\mathrm{a}}_i \rangle$$

  where arg max considers the $\hat{R}_j$ largest values.

- *List decoder*: In the list decoder we first run a non-negative least squares algorithm that gives us an estimate $\hat{\vec{b}}_j$ of $\vec{b}_j$. But this does not guarantee an

output signal either of the required sparsity or with elements strictly from the set $\{0, 1\}$ (as we know apriori from the problem). To address this, we perform a hard thresholding operation on each element of $\hat{\vec{b}}_j$ and form a list of non-negative indices $\mathcal{W}_{\text{list}} = \{i : \hat{\vec{b}}_j(i) > \eta_{Th}\}$. The value of parameter η_{Th} is chosen such that the list size is larger than T. We then implement a maximum likelihood decoder within the above list of indices to find the set of R_j indices that best explain the received vector $\vec{y_j}^{\text{p}}$ i.e.,

$$\widehat{\mathcal{W}}_j^{\text{p}} = \underset{S \subseteq \mathcal{W}_{\text{list}}, |S| = R_j}{\arg\min} ||\vec{\text{y}}_j^{\text{p}}, \sum_{i \in S} \vec{\text{a}}_i||_2^2. \tag{II.8}$$

As one can observe as we decrease the value of the threshold η_{Th} the list size increases which increases the complexity of the MMSE estimator in Eq. (II.8) whereas if we increase the value of the threshold the list size decreases and the performance worsens. Clearly for a given SNR the value of the threshold η_{Th} needs to be optimized. The CS decoder outputs the set of preamble message indices $\widehat{\mathcal{W}}_j^{\text{p}}$, where $|\widehat{\mathcal{W}}_j^{\text{p}}| = R_j$, to the channel coding decoder.

Error energy test

This component outputs positive that preamble collision did not occur if

$$\frac{1}{N_{\text{p}}} ||\vec{\text{y}}_j^{\text{p}} - \sum_{i \in \widehat{\mathcal{W}}_j^{\text{p}}} \vec{\text{a}}_i||^2 \leq (1 + P_{\text{p}}).$$

To understand the collision detection rule, consider the input to the compressed sensing decoder $\vec{\text{y}}^{\text{p}} = \mathbf{A}\vec{b} + \vec{\text{z}}^{\text{p}}$ given in Eqn. (II.7), where $\vec{b} \in \{0, 1\}^{M_{\text{p}}}$. However this is invalid if there is a collision of preamble message indices in a sub-block i.e., two users transmitting in a sub-block chose the same preamble message index. For e.g., let $R_j = 3$ and the set of preamble message indices chosen by the three users

transmitting in sub-block j be $\{1, 2, 2\}$. In this case, the compressed sensing decoder outputs a set of three distinct message indices since $R_j = 3$ which leads to an error. The idea here is that the collision detection rule prevents such cases from proceeding to further decoding with the incorrect set of preamble message indices.

Channel coding decoder

We employ joint belief propagation (BP) decoder for decoding the channel coding part of the received signal. To keep it simple we describe the decoder assuming $R_j = 2$ which can be be generalized to larger values of R_j in a straight forward manner. Without loss of generality let the two message indices be $w_1 = (w_1^{\mathrm{p}}, w_1^{\mathrm{c}})$ and $w_2 = (w_2^{\mathrm{p}}, w_2^{\mathrm{c}})$ respectively. Note that the estimates of preamble message indices $\{w_1^{\mathrm{p}}, w_2^{\mathrm{p}}\}$ are available at the channel coding decoder, output from the CS decoder. Assuming appropriate demodulation is performed before the decoding step the channel coding part of the received signal, which can be written as

$$\vec{\mathrm{y}}_j^{\,\mathrm{c}} := \vec{\mathrm{y}}_j[N_{\mathrm{p}} + 1 : N] = \sum_{i \in \{1,2\}} \pi_{\tau_{w_i^{\mathrm{p}}}}(\vec{\mathrm{c}}_{w_i^{\mathrm{c}}}) + \vec{\mathrm{z}}_j[N_{\mathrm{p}} + 1 : N_{\mathrm{c}}],$$

is input to the joint BP decoder. As we can observe, the codeword before being transmitted across the GMAC channel is permuted according to a permutation chosen as a function of the preamble message index. Therefore in the joint BP decoder we need to apply the permutations and their inverses on the messages whenever they are being sent to and from the MAC nodes respectively. The schematic of the joint Tanner Graph of the two users is shown in Fig. II.3.

Given the received signal $\vec{\mathrm{y}}_j^{\mathrm{c}}$ the joint BP decoder proceeds iteratively in a similar manner to that of a single user AWGN channel decoding apart from an extra step of messages being sent to and received from the MAC node in each iteration. We use

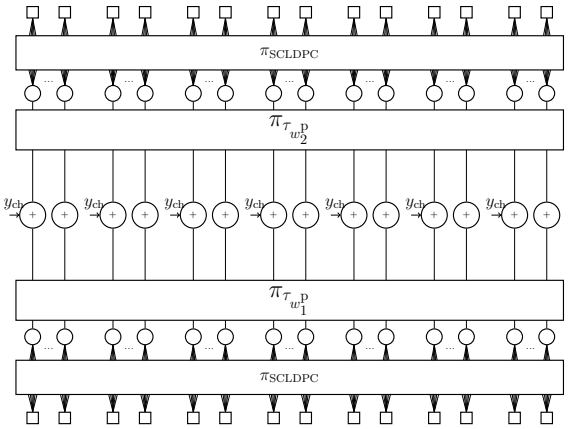

Figure II.3: Schematic showing the joint Tanner graph, of the channel coding component, for two users with message indices w_1 and w_2. In the SC-LDPC code π_{SCLDPC} refers to a random permutation of the edge connections from check nodes to bit nodes. For more details refer to [1]. We introduce multiple access (MAC) node denoting the sum over the multiple access channel. For e.g., k-th MAC node is represented by $\vec{\text{y}}_j^{\text{c}}[k] = \sum_{i\in\{1,2\}} \vec{c}_{w_i^{\text{c}}}[\pi^k_{\tau_{w_i^{\text{p}}}}] + \vec{\text{z}}_j[k]$.

the following notation for the messages passed in the joint BP decoder:

- $u^1_{i,\text{MAC}}, u^1_{i,j}$: messages passed from i-th bit node of user 1 to the corresponding MAC node and SCLDPC check node j respectively
- $v^1_{j,i}$: message passed from SCLDPC check node j to bit node i of user 1
- $v^1_{\text{MAC},i}$: message passed to i^{th} bit node from corresponding MAC node of user 1.

The messages for user 2 are defined similarly. Refer to Fig. II.4 for a graphical representation of the messages. The message passing rules in the joint message passing decoder can be summarized as following.

bit node:

$$u^1_{i,j} = v^1_{\mathrm{MAC},i} + \sum_{j' \neq j, j' \in \mathcal{N}(i)} v^1_{j',i}$$

$$u^1_{i,\mathrm{MAC}} = \sum_{j \in \mathcal{N}(i)} v^1_{j,i}$$

SCLDPC check node:

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

MAC node:

$$v^1_{\mathrm{MAC},i} = h(u^2_{i,\mathrm{MAC}}, y_{i,\mathrm{ch}}) \tag{II.9}$$

$$v^2_{i,\mathrm{MAC}} = h(u^1_{i,\mathrm{MAC}}, y_{i,\mathrm{ch}}) \quad \text{where}$$

$$h(l, y) = \log \frac{1 + e^l e^{2(y-1)/\sigma^2}}{e^l + e^{-2(y+1)/\sigma^2}}.$$

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

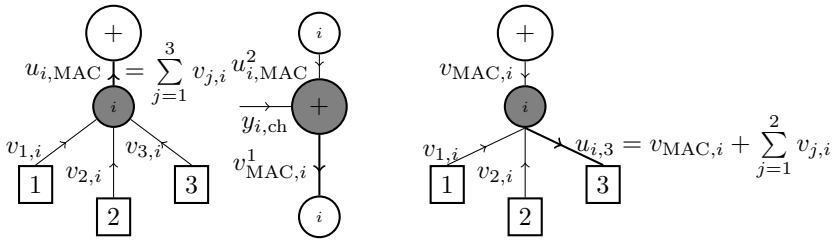

Figure II.4: Message passing rules at individual nodes on the joint Tanner graph of two users. The message passing rules at the check nodes of the SCLDPC code are identical to the single user channel coding case.

II.B.4 Decoding process across sub-blocks - SIC

For any sub-block j, if $\hat{R}_j <= T$ the T-user GMAC decoder within the sub-block, described in the previous sub-section, outputs the set of messages transmitted during the j-th sub-block. Given the decoded set of messages $\{w_i, i \in \mathcal{N}_j\}$ for sub-block j and that the preamble collision detector output is negative, for each decoded message w_i:

- the sub-blocks where the codeword is repeated can be obtained using the function $g(w_i)$,
- the codeword corresponding to message w_i is subtracted or 'peeled off' from the received signal in the corresponding repeated sub-blocks and
- in each of the repeated sub-blocks, the estimate $\hat{R}_k$ (k being the sub-block) is updated (reduced by one) 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 less than T codewords remain. The above described iterative decoding process is known in the literature as successive interference cancellation (SIC).

II.C Choice of parameters and analysis

In this section we analyze the performance of different components of the proposed scheme and the effect each of them has on the overall performance. At j-th sub-block where $R_j \leq T$ let us define the following error events:

- $\mathcal{E}_{\mathrm{p}j}$: Given there is no preamble collision, let $\mathcal{E}_{\mathrm{p}j}$ be the event that the output of the compressed sensing decoder is incorrect i.e., $\widehat{\mathcal{W}}_j^{\mathrm{p}} \neq \mathcal{W}_j^{\mathrm{p}}$. The event $\mathcal{E}_{\mathrm{p}}$ is defined for the worst case $R_j = T$

- $\mathcal{E}_{\mathrm{e}j}$: Let $\mathcal{E}_{\mathrm{e}j}$ be the event that the error energy test makes an error. With the following notations:
  - Given that there is no preamble collision and the compressed sensing decoder is correct let $\mathcal{E}^0_{\mathrm{e}j}$ be the event the error energy test detects a preamble collision and
  - let $\mathcal{E}^1_{\mathrm{e}j}$ be the event there exists a collision but the energy test fails to detect the preamble collision

  we can see that $\mathcal{E}_{\mathrm{e}j} = \mathcal{E}^0_{\mathrm{e}j} \cup \mathcal{E}^1_{\mathrm{e}j}$.

- $\mathcal{E}_{\mathrm{c}j}$: Given there is no preamble collision and that the preamble message indices are decoded successfully, let $\mathcal{E}_{\mathrm{c}j}$ be the event that the channel decoder fails to recover all the channel coding message indices correctly. The event $\mathcal{E}_{\mathrm{c}}$ is defined for the worst case, when $R_j = T$

- $\mathcal{E}_{\mathrm{SIC}}$: Let $\mathcal{E}_{\mathrm{SIC}}$ be the event that a random user is not recovered by the SIC decoding process

We observe that the overall decoding process within a given sub-block j making an error is a disjoint union of the above described events i.e.,

$$\begin{aligned}\mathcal{E}_j &= \mathcal{E}_{\mathrm{p}j} \cup \mathcal{E}^0_{\mathrm{e}j} \cup \mathcal{E}_{\mathrm{c}j} \cup \mathcal{E}^1_{\mathrm{e}j} \\ &= \mathcal{E}_{\mathrm{p}j} \cup \mathcal{E}_{\mathrm{e}j} \cup \mathcal{E}_{\mathrm{c}j}.\end{aligned}$$

The per user error probability P_e, which is equivalent to $\Pr\left(\mathcal{E}_{\text{SIC}}\right)$, can be bounded as following:

$$\begin{aligned} P_e = \Pr\left(\mathcal{E}_{\text{SIC}}\right) &\leq \Pr\left(\mathcal{E}_{\text{SIC}} | \ (\bigcup_j \mathcal{E}_j)^c\right) + \Pr\left(\bigcup_j \mathcal{E}_j\right) \\ &\leq \Pr\left(\mathcal{E}_{\text{SIC}} | \ \bigcap_j \mathcal{E}_j^c\right) + \Pr\left(\bigcup_j \mathcal{E}_{\text{p}j} \cup \mathcal{E}_{\text{e}j} \cup \mathcal{E}_{\text{c}j}\right) \\ &\leq \Pr\left(\mathcal{E}'_{\text{SIC}}\right) + \sum_j (\Pr(\mathcal{E}_{\text{p}j}) + \Pr(\mathcal{E}_{\text{e}j})\Pr(\mathcal{E}_{\text{c}j})) \\ &\leq \Pr\left(\mathcal{E}'_{\text{SIC}}\right) + V\left(\Pr(\mathcal{E}_{\text{p}}) + \Pr(\mathcal{E}_{\text{e}}) + \Pr(\mathcal{E}_{\text{c}})\right) \end{aligned} \tag{II.10}$$

where $\mathcal{E}'_{\text{SIC}}$ is the event of a user not being recovered under the SIC decoder assuming that the compressed sensing decoder, collision detector and the channel decoder do not make any errors. The precise characterization of this decoding process, referred to as simplified SIC, that can be used to evaluate $\Pr\left(\mathcal{E}'_{\text{SIC}}\right)$ is given in Def. 1. The multiplicative factor V in Eqn. (II.10) union bounds the total number of instances(sub-blocks) compressed sensing decoder, the collision detector or the channel decoder can commit an error.

Definition 1 (Simplified SIC decoder)**.** We define simplified SIC decoder as an iterative decoding process on a bipartite graph with two types of nodes, variable and slot. Consider a graph wherein the users represent variable nodes and sub-blocks represent slot nodes. Each variable node is associated with a unique preamble message index chosen independently and uniformly at random from $[M_{\text{p}}]$. Simplified SIC decoder proceeds iteratively on the bipartite graph in which at any slot node if the number of variable nodes connected is less than or equal to T:

- if there is no preamble collision between the connected variable nodes, then the respective variable nodes are assumed to have been decoded successfully. All

the connected variable nodes and their edges will be peeled off from the graph

- if there is a preamble collision, then the slot node is simply ignored in this iteration.

The idea behind the simplification is that the sub-blocks in which there is a preamble collision need not necessarily result in an error, but they can be resolved in future iterations when one of the colliding users has been decoded and peeled off from the sub-block.

II.C.1 Compressed sensing problem and design choices

In this section we discuss the choice of parameters T, B_{c} (or equivalently M_{p}), sensing matrix $\mathbf{A}$ and analyze the performance of the preamble component for various such choices under the correlation and list decoders described in Sec.II.B.3.

We consider two options for the choice of sensing matrix: (i) random matrix with each entry chosen according to Rademacher distribution, referred to as *random ensemble* and (ii) sensing matrix derived as a subset of a binary code with good minimum distance properties, referred to as *binary ensemble*.

random ensemble

For a given N_{p} and M_{p} a sensing matrix $\mathbf{A} = [a_{ij}]_{i\in[N_{\mathrm{p}}],j\in[M_{\mathrm{p}}]}$ from *random* ensemble is obtained by choosing each $a_{ij} = \pm\sqrt{P_{\mathrm{p}}}$, independently, with equal probability.

binary ensemble

A sensing matrix from *binary* ensemble is derived as a subset of a binary linear code with appropriate scaling and shifting. More precisely, for a given N_{p} and M_{p}, let $\mathcal{C}_{\mathrm{bin}}$ be a subset of size M_{p}, not necessarily a sub-code, of a binary linear code with block length N_{p}. Then the sensing matrix is obtained by $\mathbf{A} = \sqrt{P_{\mathrm{p}}}(1 - 2\mathcal{C}_{\mathrm{bin}})$.

Also let the minimum and maximum Hamming distances between any two binary vectors in $\mathcal{C}_{\text{bin}}$ be represented by $d_{\min}$ and $d_{\max}$ respectively.

For the *random* and *binary* ensembles described above the following lemma gives the probability of error under correlation decoder for a given T denoting the sparsity.

Lemma 2 (Compressed sensing with correlation decoder)**.** Consider the T-sparse support recovery problem where let $\{1, 2, \ldots, T\}$ be the set of sparse indices without loss of generality and $\vec{\mathrm{y}}$ be the preamble part of the received vector. The probability of error for the correlation decoder can then be bounded by

$$\begin{aligned} \Pr(\mathcal{E}_{\text{p}}) &= 1 - (1 - (M_{\text{p}} - T)\Pr(\mathcal{E}_{\text{corr}}))^T \\ &\leq T(M_{\text{p}} - T)\Pr(\mathcal{E}_{\text{corr}}), \end{aligned} \tag{II.11}$$

where $\Pr(\mathcal{E}_{\text{corr}})$ denotes the the probability of the error event that the correlation $\langle \vec{\mathrm{y}}, \vec{\mathrm{a}}_i \rangle \leq \langle \vec{\mathrm{y}}, \vec{\mathrm{a}}_j \rangle$ for some $i \leq T$ and $j > T$. For the *random* and *binary* ensembles this can be upper bounded by

$$\text{random ensemble:} \qquad \Pr(\mathcal{E}_{\text{corr}}) \leq \exp\left\{ \frac{-N_{\text{p}} P_{\text{p}}}{2(2 + (2T-1)P_{\text{p}})} \right\} \tag{II.12}$$

$$\text{binary ensemble:} \qquad \Pr(\mathcal{E}_{\text{corr}}) \leq \exp\left\{ \frac{-P_{\text{p}} d_{\max}(1 - T(1 - d_{\min}/d_{\max}))^2}{2} \right\}. \tag{II.13}$$

Proof. We observe that the correlation decoder makes an error if there exists $i \leq T$ and $j > T$ such that $\langle \vec{\mathrm{y}}, \vec{\mathrm{a}}_i \rangle \leq \langle \vec{\mathrm{y}}, \vec{\mathrm{a}}_j \rangle$ the probability of which is given by the right hand side in Eqn. (II.11). The analysis for the event $\mathcal{E}_{\text{corr}}$ and the bounds in Eqns. (II.12) and (II.13) are provided in Appendix. II.E.1. □

For the random ensemble we observe from Equations. (II.11) and (II.12) that

the error probability of correlation decoder decreases exponentially in the number of channel uses N_{p}. However, with respect to SNR P_{p}, the rate of decay is very slow. In fact it converges to a positive value

$$\Pr(\mathcal{E}_{\mathrm{corr}}) \to \exp\left\{\frac{-N_{\mathrm{p}}}{2(2T-1)}\right\}.$$

If we consider the binary ensemble, the error probability of the overall decoder decays exponentially in both the channel uses N_{p} and SNR P_{p} given that the subset $\mathcal{C}_{\mathrm{bin}}$ has the following properties:

- The gap between minimum and maximum Hamming distances $d_{\max} - d_{\min}$ is small
- The minimum distance $d_{\min}$ is large. Note that this in conjunction with the above condition implies a large $d_{\max}$ which is necessary for a large exponent $\frac{-P_{\mathrm{p}} d_{\max}(1-T(1-d_{\min}/d_{\max}))^2}{2}$

Based on the design objectives outlined above, we design a sensing matrix from the *binary ensemble* for a toy example with parameters $M_{\mathrm{p}} = 512, N_{\mathrm{p}} = 63$.

Example 3 (Sensing matrix from binary ensemble)**.** Let binary code $\mathcal{C}_{\mathrm{BCH}}$ be the BCH$(63, 10)$ code of size 1024. We obtain a subset of $\mathcal{C}_{\mathrm{BCH}}$ of size M_{p} by the following decomposition

$$\mathcal{C}_{\mathrm{BCH}} = \mathcal{C}_0 \cup \mathcal{C}_1 \quad \text{such that } c \in \mathcal{C}_1 \iff \bar{c} \in \mathcal{C}_0,$$

where $\bar{c} = \mathbf{1} \oplus c$ i.e., the one's complement of c. We choose the sensing matrix of size $N_{\mathrm{p}} \times M_{\mathrm{p}} = 63 \times 512$ as $\mathbf{A} = [\vec{\mathrm{a}}_1, \vec{\mathrm{a}}_2, \ldots, \vec{\mathrm{a}}_{M_{\mathrm{p}}}]$, where $\vec{\mathrm{a}}_i = \sqrt{P_{\mathrm{p}}}(1 - 2\vec{\mathrm{c}}_i), \vec{c}_i \in \mathcal{C}_1$, i.e., $a_{ij} \in \{-\sqrt{P_{\mathrm{p}}}, \sqrt{P_{\mathrm{p}}}\} \forall i, j$. This specific decomposition allows us to maintain the

minimum distance $d_{\min} = 28$ identical to the original code $\mathcal{C}_{\text{BCH}}$ while reducing the maximum distance for the subset $\mathcal{C}_1$ to $d_{\max} = 36$ from 63 of the original code.

In Fig. II.5 we present the error performance results for the sensing matrix in Example 3 under correlation decoder and compare with the performance of a sensing matrix from the random ensemble. It can be clearly seen that correlation decoder is sub-optimal. Therefore we also present the performance of both the ensembles under the list decoder. Although the list decoder is difficult to analyze primarily due to the LASSO and constrained least squares optimization algorithm components, it can be seen from Fig. II.5 that the list decoder has superior performance when compared to the correlation decoder via numerical simulations.

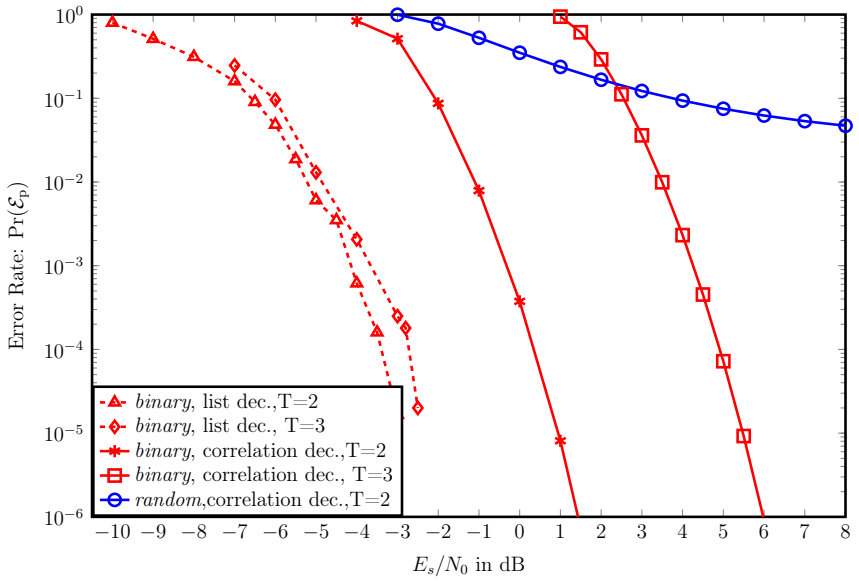

Figure II.5: For $N_{\text{p}} = 63, M_{\text{p}} = 512$ we compare the performance of the *binary* and *random* ensembles under list and correlation decoders for $T = \{2, 3\}$. The sensing matrix for binary ensemble is given in Ex. 3. For the correlation decoder we simply use the performance bounds given in Lemma. 2 whereas for the list decoder we perform numerical simulations using list decoder where we use non-negative least squares for the first component of the decoder.

II.C.2 Energy test

In this section we analyze the the performance of the error energy test component.

Lemma 4. The probability of the event that the energy test makes an error can be bounded by

$$\Pr(\mathcal{E}_{\mathrm{e}}) \leq \left((1+P_{\mathrm{p}})e^{-P_{\mathrm{p}}}\right)^{N_{\mathrm{p}}/2} + \frac{T(T-1)}{2M_{\mathrm{p}}}\Pr\left(\frac{1}{N_{\mathrm{p}}}||\vec{\mathrm{a}}_i + \vec{\mathrm{a}}_j + \vec{\mathrm{z}}||^2 \leq (1+P_{\mathrm{p}})\right),$$

where i, j are distinct indices chosen randomly from the set $[M_{\mathrm{p}}]$.

Proof. If we let $\Pr(\mathcal{E}_{\mathrm{coll}})$ be the event that there is a preamble collision, then

$$\begin{aligned}
\Pr(\mathcal{E}_{\mathrm{e}}) &= \Pr(\mathcal{E}_{\mathrm{e}}, \mathcal{E}_{\mathrm{coll}}^c) + \Pr(\mathcal{E}_{\mathrm{e}}, \mathcal{E}_{\mathrm{coll}}) \\
&\leq \Pr(\mathcal{E}_{\mathrm{e}}|\mathcal{E}_{\mathrm{coll}}^c) + \Pr(\mathcal{E}_{\mathrm{coll}})\Pr(\mathcal{E}_{\mathrm{e}}|\mathcal{E}_{\mathrm{coll}}) \\
&\overset{(a)}{=} \Pr(\mathcal{E}_{\mathrm{e}}^0) + \Pr(\mathcal{E}_{\mathrm{coll}})\Pr(\mathcal{E}_{\mathrm{e}}^1)
\end{aligned}$$

where substituting the results from Lemmas. 5, 6 and 7 in (a) completes the proof. □

Lemma 5. $\Pr(\mathcal{E}_{\mathrm{e}}^0) \leq \left((1+P_{\mathrm{p}})e^{-P_{\mathrm{p}}}\right)^{N_{\mathrm{p}}/2}$

Proof. Given there is no preamble collision and the compressed sensing decoder is successful, $\Pr(\mathcal{E}_{\mathrm{e}}^0)$ can be bounded as

$$\begin{aligned}
\Pr(\mathcal{E}_{\mathrm{e}}^0) &= \Pr\left(\frac{1}{N_{\mathrm{p}}}||\vec{\mathrm{y}} - \sum_{i\in\widehat{\mathcal{N}}_j}\vec{\mathrm{a}}_i||^2 > 1+P_{\mathrm{p}}\right) \\
&= \Pr\left(\frac{1}{N_{\mathrm{p}}}||\vec{\mathrm{z}}||^2 > 1+P_{\mathrm{p}}\right)
\end{aligned}$$

the probability of which can be upper bounded using the tail bound of chi-squared distribution. □

Corollary 6. Let i, j be distinct indices chosen randomly from the set $[M_{\mathrm{p}}]$, then

$$\Pr(\mathcal{E}_{\mathrm{e}}^{1}) \leq \Pr\left(\frac{1}{N_{\mathrm{p}}}||\vec{\mathrm{a}}_i + \vec{\mathrm{a}}_j + \vec{\mathrm{z}}||^2 \leq 1 + P_{\mathrm{p}}\right)$$

Proof. Let the set of T preamble indices be $\mathcal{W}^{\mathrm{p}}$ and the output of the compressed sensing decoder be $\widehat{\mathcal{W}}^{\mathrm{p}}$.

$$\Pr(\mathcal{E}_{\mathrm{e}}^{1}) = \Pr\left(\frac{1}{N_{\mathrm{p}}}||\vec{\mathrm{y}}_j^{\mathrm{p}} - \sum_{i \in \widehat{\mathcal{W}}_j^{\mathrm{p}}} \vec{\mathrm{a}}_i||^2 \leq (1 + P_{\mathrm{p}})\right)$$

$$\begin{aligned}
&= \Pr\left(\frac{1}{N_{\mathrm{p}}}||\sum_{i \in \mathcal{W}_j^{\mathrm{p}}} \vec{\mathrm{a}}_i - \sum_{i \in \widehat{\mathcal{W}}_j^{\mathrm{p}}} \vec{\mathrm{a}}_i + \vec{\mathrm{z}}||^2 \leq (1 + P_{\mathrm{p}})\right) \\
&= \Pr\left(\frac{1}{N_{\mathrm{p}}}||\sum_{i \in \mathcal{W}^{\mathrm{p}} \bigtriangleup \widehat{\mathcal{W}}^{\mathrm{p}}} \vec{\mathrm{a}}_i + \vec{\mathrm{z}}||^2 \leq (1 + P_{\mathrm{p}})\right) \\
&\overset{(b)}{\leq} \Pr\left(\frac{1}{N_{\mathrm{p}}}||\vec{\mathrm{a}}_i + \vec{\mathrm{a}}_j + \vec{\mathrm{z}}||^2 \leq (1 + P_{\mathrm{p}})\right)
\end{aligned}$$

where for (b) we recall that the compressed sensing decoder outputs T distinct preamble indices whereas $\mathcal{W}^{\mathrm{p}}$ has atleast one repeating preamble index and thus $|\mathcal{W}^{\mathrm{p}} \bigtriangleup \widehat{\mathcal{W}}^{\mathrm{p}}| \geq 2$. □

Lemma 7. $\Pr\left(\mathcal{E}_{\mathrm{coll}}\right) \leq \frac{T(T-1)}{2M_{\mathrm{p}}}$.

Proof. Let us consider the event $\mathcal{E}_{\mathrm{coll}}^{c}$ where the T users in the slot picked a unique preamble message index. Note that in total there are M_{p} possible preamble indices

for each user.

$$\begin{aligned} \Pr(\mathcal{E}_{\text{coll}}^c) &= \frac{M_{\text{p}}(M_{\text{p}}-1)\dots(M_{\text{p}}-(T-1))}{M_{\text{p}}^T} \\ \implies \Pr(\mathcal{E}_{\text{coll}}) &= 1-\prod_{i=0}^{T-1}(1-\frac{i}{M_{\text{p}}}) \\ &\leq \frac{T(T-1)}{2M_{\text{p}}}. \end{aligned} \tag{II.14}$$

□

II.C.3 Channel coding problem

In the following subsection we will look at the analysis of the T-GMAC channel coding problem and the bounds on performance. Although the information theoretic limits for the multiple access problem especially the symmetric rate region are well known these do not prove to be very useful for our purposes. It is because even though the block lengths we are interested in are considerably large the information length (or equivalently the code size for each user) is small. Therefore we will be considering the finite length performance especially we will use the finite length random coding bounds for the Gaussian multiple access channel derived by Polyanskiy [2]. The following lemma is identical to Thm. 1 in [2] except for the difference that we are interested in the case where error is declared if atleast one of the users messages is not in the decoded set (see event $\mathcal{E}_{3j}$) in contrast to [2] where the error probability is defined similar to Eqn. (II.2).

Lemma 8. There exists an (N', M_1) random-access code for T-user satisfying the power constraint P (see Eqn. (II.4)) with the probability of error under maximum-

likelihood decoder bounded by

$$P(\mathcal{E}_3) \leq \mathsf{h}_{\mathrm{FBL}}(N', M_{\mathrm{c}}, T, P) := \sum_{t=1}^{T} \min(p_t, q_t) + p_0, \tag{II.15}$$

where

$$\begin{aligned}
p_0 &= \frac{\binom{T}{2}}{M_{\mathrm{c}}} + T\Pr\left(\sum_{j=1}^{N'} Z_j^2 > \frac{N'P}{P'}\right) \\
p_t &= e^{-N'E(t)} \\
E(t) &= \max_{0\leq\rho,\rho_1\leq 1} -\rho\rho_1 t R_1 - \rho_1 R_2 + E_0(\rho, \rho_1) \\
E_0 &= \rho_1 a + \frac{1}{2}\log(1 - 2b\rho_1) \\
a &= \frac{\rho}{2}\log(1 + 2P't\lambda) + \frac{1}{2}\log(1 + 2P't\mu) \\
b &= \rho\lambda - \frac{\mu}{1 + 2P't\mu}, \mu = \frac{\rho\lambda}{1 + 2P't\lambda} \\
\lambda &= \frac{P't - 1 + \sqrt{D}}{4(1 + \rho_1\rho)P't} \\
D &= (P't - 1)^2 + 4P't\frac{1 + \rho\rho_1}{1 + \rho} \\
R_1 &= \frac{1}{N'}\log M_{\mathrm{c}} - \frac{1}{N'}\log(t!)
\end{aligned}$$

$$\begin{aligned}
R_2 &= \frac{1}{N'}\log\binom{T}{t} \\
q_t &= \inf_{\gamma} \Pr[I_t \leq \gamma] + \exp\{N'(R_1 + R_2) - \gamma\}.
\end{aligned}$$

and

$$
\begin{aligned}
I_t &= \min_{|S_0|=t, S_0 \subseteq [T]} N' C_t + \frac{\log e}{2} \left(\frac{|| \sum_{i \in S_0} \vec{c}_i + \vec{z}||_2^2}{1 + P't} - ||\vec{z}||_2^2 \right) \\
C_t &= \frac{1}{2} \log(1 + P't) \\
\vec{z} &\sim \mathcal{N}(0, \mathbf{I}_{N'}).
\end{aligned}
$$

Proof. In [2], author Y. Polyanskiy considers the T-user GMAC problem with power constraint P according to Eqn. (II.4). Let W be the set of messages of size T, chosen by the users uniformly without replacement and $\hat{W}$ be the set of messages of size T output by the decoder. The author considers a random Gaussian codebook generated from Gaussian process $\mathcal{N}(0, P'\mathbf{I}_n)$, $(P' < P)$, and maximum-likelihood decoder and shows that

$$
\Pr(|W \backslash \hat{W}| = t) = \min(p_t, q_t). \tag{II.16}
$$

It was also shown that p_0 is the total variation distance of a random variable of maximum value 1 when the measure under which a) the messages are sampled *independently* rather than *without replacement* and b) the codeword is set to zero-vector if the total power of the random codeword is larger than nP is replaced by the measure considered in showing Eqn. (II.16) i.e., messages sampled independently and disregarding the strict power constraint on each codeword. These results along with the observation that

$$
\begin{aligned}
\Pr(\mathcal{E}_3) &= 1 - \prod_{i=1}^{t} \Big(1 - \Pr(\hat{w}_i \notin \hat{W}) \Big) + p_0 \\
&\leq \sum_{t=1}^{T} \Pr(|S \backslash \hat{S}| = t)
\end{aligned}
$$

completes the proof. □

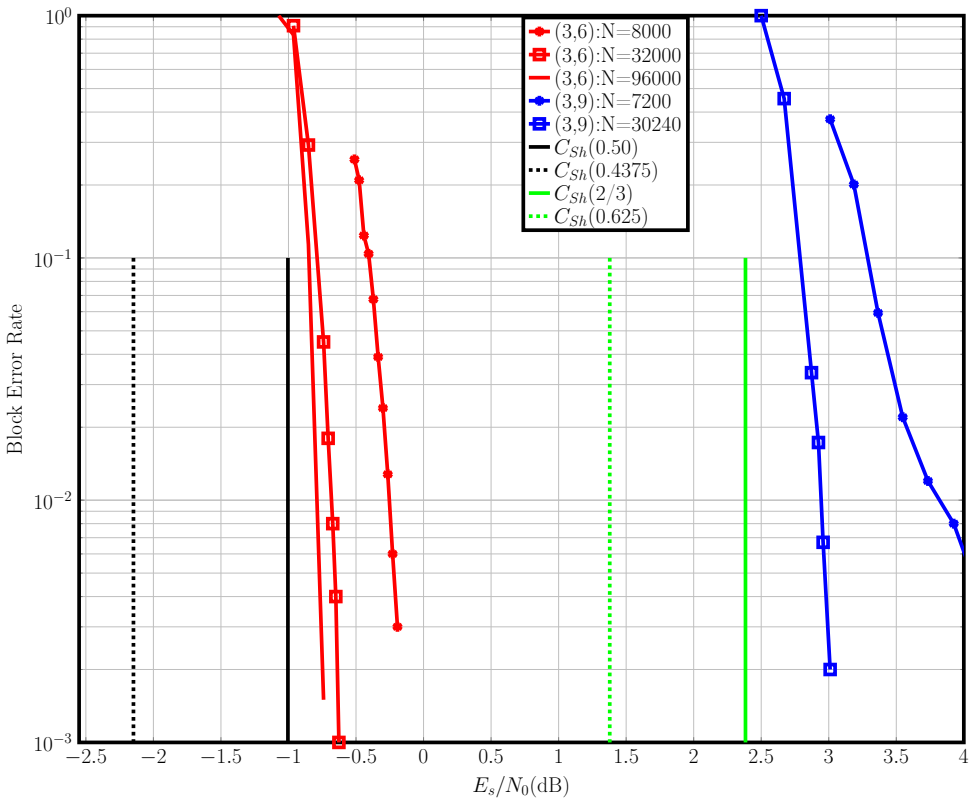

Figure II.6: We simulate the performance of the channel coding component alone using regular $(3, 6)$ and $(3, 9)$ spatially-coupled LDPC (SC-LDPC) ensembles for increasing block lengths for two user Gaussian MAC channel. The results demonstrate that it is possible to achieve the capacity of two-user GMAC (and can be generalized for T-GMAC) using identical code books at all the users.

II.C.4 Successive interference cancellation

In the channel coding literature for LDPC codes on binary erasure channel and sparse signals via Tanner graphs literature the symmetric interference cancellation is traditionally studied under the name of peeling decoder which is an iterative process in which if a right node (slot in our case) is connected to only one left node (user)

the corresponding left node and all its connections are peeled off from the bipartite graph. This is essentially the symmetric interference cancellation process described in Sec. II.B.4 except that we peel off the connections from a right node if the number of variable nodes connected is less than or equal to T instead of 1. Although density evolution methods are well studied to predict the performance of such decoding processes all the existing density evolution methods are for values of $T = 1$. Before we address this issue let us define the considered peeling process precisely.

Definition 9 (T-peeling)**.** We define an ideal SIC decoder as the decoder in which at each slot, if the number of users transmitted and are still undecoded is less than or equal to T, then the remaining undecoded users in that slot are decoded with zero error. In other words in the ideal SIC process there are no hash collisions in any slot and the channel and sparse signal decoders are assumed to be zero-error. This process proceeds iteratively until all the users are decoded or there are no slots with undecoded users less than or equal to T. We also refer to this as T-peeling process.

Lemma 10 (Density Evolution (DE))**.** Let the left and right degree distributions (d.d.) of the bipartite graph from the edge perspective be $\lambda(x)$ and $\rho(x)$. Then let x_t be the probability that an edge in the graph, in iteration t of the T-peeling process, is connected to a left node that is undecoded yet. Then the recurrence relation for x_t corresponding to the T-peeling process is given by

$$y_t = \left[\sum_{r=1}^{T} \rho_r + \sum_{r>T} \rho_r \left(\sum_{t=0}^{T-1} \binom{r-1}{t} (1-x_t)^{r-1-t} x_t^t\right)\right], \tag{II.17}$$

$$x_{t+1} = \lambda(1-y_t). \tag{II.18}$$

Proof. Proof is provided in Appendix II.E.2. □

Let $L(x) = \sum_{i=1}^{l_{\max}} L_i x^i$ be the left d.d according to which the the users choose their repetition parameters as described in Sec. II.B.1 i.e., $\Pr(L_w = i) = L_i$. Also let the average left degree of this distribution be $l_{\text{avg}} = \sum_i i L_i$. Then according to our transmission policy the right d.d. $R(x)$ is Binomial distributed with parameters $(K_{\text{a}} l_{\text{avg}}, 1/V)$ and in the limit $K_{\text{a}} \to \infty$ $R(x)$ can be approximated as Poisson distribution with parameter $r_{\text{avg}} = \frac{K_{\text{a}} l_{\text{avg}}}{V}$. Thus, asymptotically in K_{a}, it can be seen that $R(x) = e^{-r_{\text{avg}}(1-x)}$ and $\rho(x) = R'(x)/R'(1) = e^{-r_{\text{avg}}(1-x)}$. For more details refer to [9].

Lemma 11. For $V = \alpha K_a$ where α is fixed the asymptotic performance of our transmission scheme under the ideal SIC decoding process can be characterized by

$$\lim_{K_{\text{a}} \to \infty} \Pr(\mathcal{E}_{\text{SIC}}(K_{\text{a}}, T)) = L(1 - y_\infty)$$
$$\text{where } y_\infty = \lim_{t \to \infty} y_t,$$

and $\Pr(\mathcal{E}_{\text{SIC}}(K_{\text{a}}, T))$ is the probability that the ideal SIC process does not recover a user given there are K_a users. Here the initial condition is $x_0 = 1$ and the evolution of x_t, y_t is given by the DE relationship in Lem. 10.

As we can see from Eqns. (II.18) and (II.17) that $x_t = 0$ is a fixed point if and only if $\lambda_0 = 0$. This leads us to the following result characterizing he threshold behavior of the system.

Definition 12 (Density Evolution Threshold)**.** If $L_1 = 0$ we define the density evolution threshold α^*_{DE} to be

$$\alpha^*_{\text{DE}} \triangleq \inf\{\alpha : \lim_{K_{\text{a}} \to \infty} \Pr(\mathcal{E}_{\text{SIC}}(K_{\text{a}}, T)) = 0\}.$$

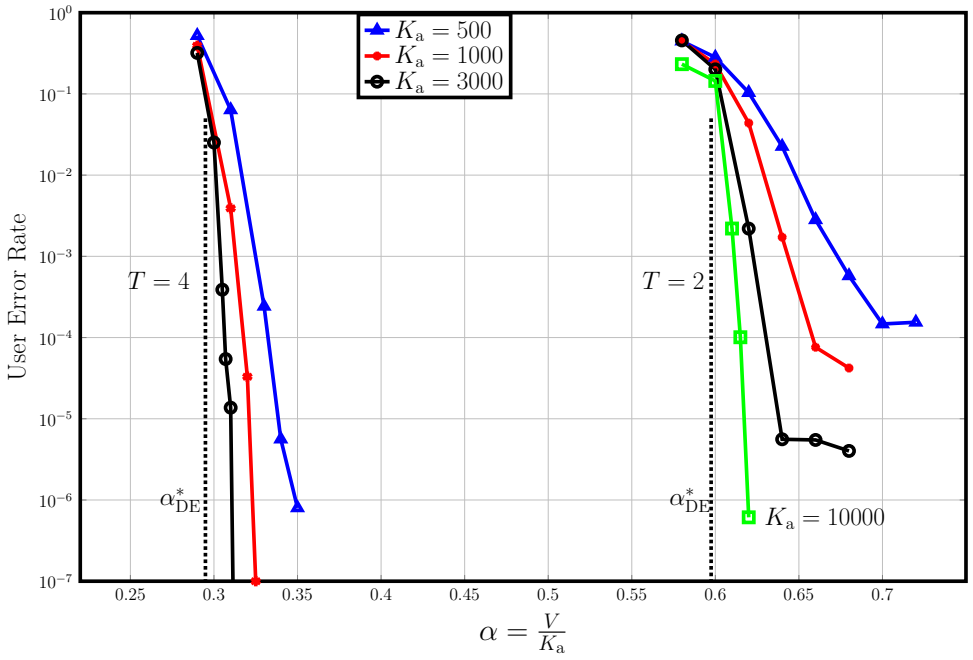

Figure II.7: α^*_{DE} is the density evolution threshold computed for $L(x) = x^2$ and $T = \{2, 4\}$ from Lemmma. (10). We validate the threshold behavior by evaluating the T-peeling performance via Monte Carlo simulations for increasing blocklengths. We observe that the simulations indeed confirm the threshold behavior for values of α above the DE threshold.

We validate the threshold behavior via simulations. For a fixed left d.d $L(x) = x^2$ we first compute the density evolution thresholds according to Def. 12 to be 0.5975 and 0.2949 for $T = 2, 4$ respectively. We then perform Monte Carlo simulations where each time a random graph is chosen as described in Sec. II.B.1 for increasing values of K_{a} and plot the performance as we incrase the number of slots. The results are presented in Fig. II.7. In both the cases the threshold behavior can be clearly seen that as K_{a} increases the probability of a user not being decoded decreases sharply for values of $\alpha > \alpha^*_{\text{DE}}$ and remains fairly constant for values of $\alpha \leq \alpha^*_{\text{DE}}$.

II.D Numerical results

In this section we numerically evaluate the overall performance of the proposed scheme and compare with other multiple access schemes available in the literature. In [10] apart from proposing a low complexity coding scheme for the unsourced GMAC channel the authors Ordentlich and Polyanskiy also evaluate the performance of their proposed scheme by computing the minimum SNR required to achieve the target error probability for a fixed set of parameters. To make the comparison convenient we pick identical parameters, summarized as following:

- number of bits each user intends to transmit $B = 100$
- total number of channel uses $\tilde{N} = 30,000$
- number of active users $K_{\mathrm{a}} \in [25 : 300]$
- maximum per user error probability $P_e \leq \epsilon = 0.05$.

With the parameters $B, \tilde{N}, K_{\mathrm{a}}, \epsilon$ fixed, the choices for the design parameters are as following:

1. Maximum number of users to be jointly decoded at a slot $T \in \{2, 4, 5\}$.
2. The left d.d is chosen to be $L(x) = \beta x + (1 - \beta)x^2$ (see Remark 13). The free parameter is optimized over the set $\beta \in \{0, 0.1, \ldots, 1\}$.
3. Number of preamble and channel coding message bits: $B_{\mathrm{p}} = 9,\ B_{\mathrm{c}} = B - B_{\mathrm{p}} = 91$.
4. *Sensing matrix for preamble component*: Note that $M_{\mathrm{p}} = 2^{B_{\mathrm{p}}} = 512$ is the size of the sensing matrix $\mathbf{A}$. We choose the sensing matrix of dimensions $N_{\mathrm{p}} \times M_{\mathrm{p}} = 63 \times 512$ as described in Ex. 3.

5. *Channel coding component*: The number of channel uses available for channel coding N_{c} is dependent on N which in turn depends on the total number of sub-blocks V. It is impractical to build a channel code for various rates $R_{\mathrm{c}} = \frac{B_{\mathrm{c}}}{N_{\mathrm{c}}}$ (although B_{c} is fixed, N_{c} needs to be optimized over) and evaluate the performance numerically for each set of parameters $(N_{\mathrm{c}}, B_{\mathrm{c}})$. Therefore to evaluate the performance of the channel coding component we use the finite block length achievability bound in Eqn. (II.16) due to Polyanskiy. This seems a reasonable choice as we demonstrated in Fig. xx that one can construct LDPC codes even for moderate block lengths that perform close to the above mentioned bound.

From Eqn. II.10 we want $\Pr(\mathcal{E}'_{\mathrm{SIC}}) + V\left(\Pr(\mathcal{E}_{\mathrm{p}}) + \Pr(\mathcal{E}_{\mathrm{e}}) + \Pr(\mathcal{E}_{\mathrm{c}})\right) \leq \epsilon = 0.05$. Therefore we set the target error probabilities for the individual events as $\Pr(\mathcal{E}_i) \leq \epsilon_0/3/V$, $i \in \{\mathrm{p}, \mathrm{e}, \mathrm{c}\}$ where we choose $\epsilon_0 = 0.01$ and $\Pr(\mathcal{E}'_{\mathrm{SIC}}) \leq (\epsilon - \epsilon_0) = 0.04$. For a fixed T the performance of the overall scheme i.e., the minimum E_b/N_0 required for achieving $P_e \leq \epsilon$ is computed as following:

$$\frac{E_b}{N_0} = \min_{\beta} \frac{(2-\beta)(N_{\mathrm{p}}P_{\mathrm{p}} + N_{\mathrm{c}}P_{\mathrm{c}})}{2B} \tag{II.19}$$

where

$$P_{\mathrm{p}} := \arg\min_{P} \ \max\left(\Pr(\mathcal{E}_{\mathrm{p}}), \Pr(\mathcal{E}_{\mathrm{e}})\right) \leq \frac{\epsilon_0}{3V} \tag{II.20}$$

$$P_{\mathrm{c}} := \arg\min_{P} \ \mathsf{h}_{\mathrm{FBL}}(N, B_{\mathrm{c}}, T, P)) \leq \frac{\epsilon_0}{3V} \qquad \text{(see Eqn. (II.15))} \tag{II.21}$$

$$N := \left\lfloor \frac{\tilde{N}}{V} \right\rfloor$$

$$V := \arg\min_{V} \ \Pr(\mathcal{E}'_{\mathrm{SIC}}(K_{\mathrm{a}}, V, T)) \leq \epsilon - \epsilon_0. \tag{II.22}$$

A remark on how we compute the quantities in Eqns. (II.20), (II.21) and (II.22):

- $\Pr(\mathcal{E}_{\mathrm{p}})$: We choose T preamble message indices, randomly, without replacement from the available M_{p} indices and form the measurement vector. We then use the list decoder as described in Sec. II.B.3. The probability of error $\Pr(\mathcal{E}_{\mathrm{p}})$ in Eqn. (II.20) is then computed from atleast 10^5 Monte Carlo simulations

- $\Pr(\mathcal{E}_{\mathrm{c}})$: We use the upper bound for $\Pr(\mathcal{E}_{\mathrm{c}})$ given in Lem. 4 except for the term

$$\Pr\left(\frac{1}{N_{\mathrm{p}}}||\vec{\mathrm{a}}_i + \vec{\mathrm{a}}_j + \vec{\mathrm{z}}||^2 \leq (1+P_{\mathrm{p}})\right)$$

  which we evaluate numerically from atleast 10^5 Monte Carlo simulations. In each simulation we choose indices i, j randomly without replacement from the set $[M_{\mathrm{p}}]$ and $\vec{z} \sim \mathcal{N}(0, \mathbf{I}_{N_{\mathrm{p}}})$.

- $\Pr(\mathcal{E}'_{\mathrm{SIC}}(K_{\mathrm{a}}, V, T))$: We rely on Monte Carlo simulations wherein for each simulation we generate a bipartite graph of K_{a} variable nodes and V slot nodes with edge connections as described in Sec. II.B.1. We run the simplified SIC decoder on just the bipartite graph as described in Def. 1 and evaluate the per user error probability.

Finally the results for the minimum SNR required to achieve the target error probability optimized according to Eqn. (II.19) are presented in Fig. II.8.

In Fig. II.8 the curves labelled $T = 2$, $T = 4$ and $T = 5$ correspond to the performance of our proposed scheme evaluated as described above for various values of parameter T. The curve labelled 4-fold ALOHA is the performance of the 4-fold ALOHA scheme from [10]. It can be seen that for large values of K_{a}, our proposed scheme with $T = 4$ or 5 substantially outperforms the 4-fold ALOHA and this gain is due to the iterative decoding process that is absent in 4-fold ALOHA. The curve

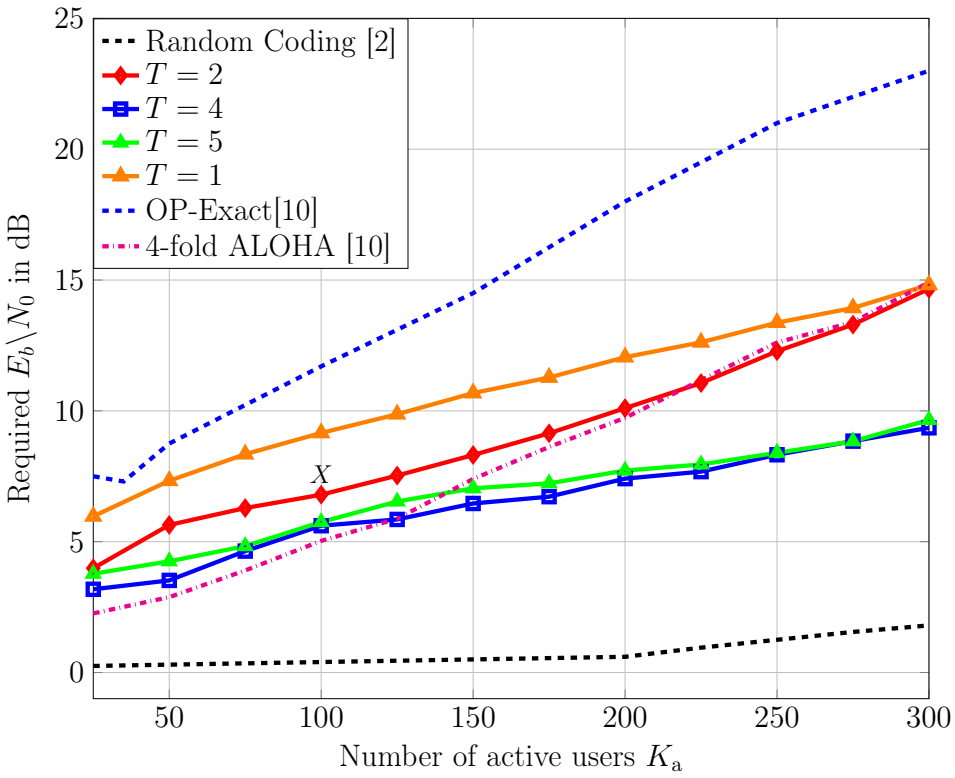

Figure II.8: Minimum E_b/N_0 required to achieve $P_e \leq 0.05$ as a function of number of users. The x mark represents the performance of our proposed scheme where for the channel coding part instead of the finite blocklength bounds given in [2], we use numerical simulation results from a regular LDPC code.

labelled OP-Exact is a reproduction of the results from [10] of the practical scheme introduced there.

The x mark represents our proposed scheme where for the channel coding part instead of the FBL bounds we use the actual simulation results. We use a rate-1/4 $(364, 91)$ LDPC code obtained from repeating every coded bit of (3,6) LDPC code twice and a message passing decoder for $T = 2$. It can be seen that the simulation results with the $(3, 6)$ LDPC code are only 0.5 dB away from the curve corresponding to $T = 2$ showing that the pragmatic coding scheme can perform close to the finite length bounds. It can also be seen that our proposed scheme provides substantial

gain over the results in [10].

In the proposed encoding scheme, for $L(x) = \beta x + (1-\beta)x^2$ each user may transmit once or twice depending on the message index chosen. We need to point out that the power constraint employed is an average over all the message indices i.e

$$\mathbb{E}_w \left[||\vec{c}_w||^2 \right] = (2-\beta)P.$$

We also present the results when the power constraint is uniform across all the codewords in the code i.e., $||\vec{c}_w||^2 \leq P \forall w$ in Fig. II.9. We achieve this, for each value of SNR E_s/N_0, by choosing $\beta = 0($ or $1)$ which in turn guarantees each codeword is repeated exactly twice (or once) irrespective of the message.

Remark 13. Although in Sec. II.C.4 we remarked that if the minimum left degree is one then zero is not a fixed point for the DE equations or in other words, in the asymptotic regime, we will have error floors rather than threshold behavior. But the effects of a minimum left degree of one in the finite number of users regime are not very clear.

II.E Appendix

II.E.1 Proof of Lem. 2

To complete the proof of Lemma. 2 we need to show that

$$\begin{aligned} \text{random ensemble:} \qquad & \Pr(\mathcal{E}_{\text{corr}}) \leq \exp\left\{ \frac{-N_{\text{p}} P_{\text{p}}}{2(2+(2T-1)P_{\text{p}})} \right\} \\ \text{binary ensemble:} \qquad & \Pr(\mathcal{E}_{\text{corr}}) \leq \exp\left\{ \frac{-P_{\text{p}}(d_{\max} - T(d_{\max} - d_{\min}))^2}{2d_{\max}} \right\}. \end{aligned}$$

According to the hypothesis in Lemma. 2 the preamble part of the received vector is written as $\vec{\text{y}} = \sum_{i=1}^{T} \vec{\text{a}}_i + \vec{\text{z}}$ where $\vec{\text{z}} \sim \mathcal{N}(0, \mathbf{I}_{N_{\text{p}}})$. For a fixed j chosen randomly

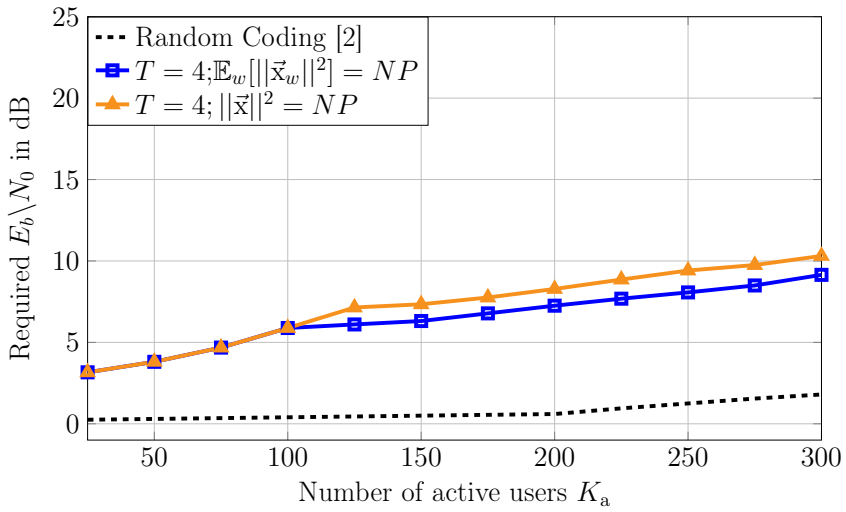

Figure II.9: Minimum E_b/N_0 required to achieve $P_e \leq 0.05$ as a function of number of users. We present the performance comparison of the average power constraint versus the uniform power constraint case. For the uniform power constraint case, number of times a codeword is repeated is constant and is independent of the message index thus resulting in equal energy being expended for all the codewords uniformly.

from $j \in \{T+1, .., M_p\}$,

$$\Pr(\mathcal{E}_{\text{corr}}) \triangleq \Pr(\langle \vec{y},\ \vec{a}_j \rangle > \langle \vec{y},\ \vec{a}_1 \rangle) \tag{II.23}$$

$$= \Pr(\langle \vec{z} + \sum_{i=2}^{T} \vec{a}_i, \vec{a}_j - \vec{a}_1 \rangle + \langle \vec{a}_1, \vec{a}_j \rangle > N_p P_p) \tag{II.24}$$

$$= \Pr\left(\frac{1}{N_p P_p} \left[\langle \vec{z}, \vec{a}_j - \vec{a}_1 \rangle + \langle \sum_{i=1}^{T} \vec{a}_i, \vec{a}_j \rangle - \langle \sum_{i=2}^{T} \vec{a}_i, \vec{a}_j \rangle \right] > 1 \right) \tag{II.25}$$

where we use the fact $||\vec{a}_i||^2 = N_p P_p \forall i$ in Eqn. (II.24).

random ensemble

For the random ensemble $\frac{1}{P_p} a_{ik} a_{jk}$ is a Rademacher random variable $\forall i \neq j, k \in [N_p]$ and thus from central limit theorem $\frac{1}{N_p P_p} \langle \vec{a}_i, \vec{a}_j \rangle \frac{1}{N_p P_p} \sum_k a_{ik} a_{jk} \rightarrow \mathcal{N}(0, \frac{1}{Np})$ asymptotically in N_p. Similarly for all $k \in [N_p]$, $\frac{1}{P_p} z_k a_{ik} \sim \mathcal{N}(0, \frac{1}{P_p})$ and $\frac{1}{N_p P_p} \sum_k z_k a_{ik} \sim$

$\mathcal{N}(0, \frac{1}{N_\mathrm{p}P_\mathrm{p}})$. Thus the right hand side in Eqn. (II.25) can be approximated as

$$\begin{aligned}
\Pr(\mathcal{E}_{\mathrm{corr}}) &= \Pr\left(\frac{1}{N_\mathrm{p}P_\mathrm{p}}\left[\langle\vec{\mathrm{z}},\vec{\mathrm{a}}_j-\vec{\mathrm{a}}_1\rangle+\langle\sum_{i=1}^{T}\vec{\mathrm{a}}_i,\vec{\mathrm{a}}_j\rangle-\langle\sum_{i=2}^{T}\vec{\mathrm{a}}_i,\vec{\mathrm{a}}_j\rangle\right]>1\right)\\
&\approx \Pr(z_{eq}+a_{eq}>1)\quad \text{where } z_{eq}\sim\mathcal{N}(0,\frac{2}{N_\mathrm{p}P_\mathrm{p}}), a_{eq}\sim\mathcal{N}(0,\frac{2T-1}{N_\mathrm{p}})\\
&\leq \exp\left\{-\frac{N_\mathrm{p}P_\mathrm{p}}{2(2+P_\mathrm{p}(2T-1))}\right\}.
\end{aligned}$$

binary ensemble

For the binary ensemble the correlation between any two vectors is bounded as $N_\mathrm{p}-2d_{\max}\leq\frac{1}{P_\mathrm{p}}\langle\vec{\mathrm{a}}_i,\vec{\mathrm{a}}_j\rangle\leq N_\mathrm{p}-2d_{\min}$. Thus the right hand side in Eqn. (II.25) can be upper bounded as

$$\begin{aligned}
\Pr(\mathcal{E}_{\mathrm{corr}}) &= \Pr\left(\left[\langle\vec{\mathrm{z}},\vec{\mathrm{a}}_j-\vec{\mathrm{a}}_1\rangle+\langle\sum_{i=1}^{T}\vec{\mathrm{a}}_i,\vec{\mathrm{a}}_j\rangle-\langle\sum_{i=2}^{T}\vec{\mathrm{a}}_i,\vec{\mathrm{a}}_j\rangle\right]>N_\mathrm{p}P_\mathrm{p}\right)\\
&\leq \Pr\left([\langle\vec{\mathrm{z}},\vec{\mathrm{a}}_j-\vec{\mathrm{a}}_1\rangle+P_\mathrm{p}T(N_\mathrm{p}-2d_{\min})-P_\mathrm{p}(T-1)(N_\mathrm{p}-2d_{\max})]>N_\mathrm{p}P_\mathrm{p}\right)\\
&= \Pr\left(z'_{eq}>2P_\mathrm{p}(d_{\max}-T(d_{\max}-d_{\min})\right)\\
&\leq \exp\left\{\frac{-P_\mathrm{p}(d_{\max}-T(d_{\max}-d_{\min}))^2}{2d_{\max}}\right\},
\end{aligned}$$

where $z'_{eq}=\langle\vec{\mathrm{z}},\vec{\mathrm{a}}_j-\vec{\mathrm{a}}_1\rangle\sim\mathcal{N}(0,4P_\mathrm{p}d_{1j})$, d_{1j} is the Hamming distance between the vectors $\vec{\mathrm{a}}_1$ and $\vec{\mathrm{a}}_j$. Thus the maximum variance z'_{eq} can have is $4P_\mathrm{p}d_{\max}$ which when used in the tail bound for normal distribution gives the required upper bound.

II.E.2 Proof of Lem. 10

In the context of low density parity check (LDPC) codes 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 binary erasure channel where each bit is erased with probability ϵ, 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 [22]

$$y_t = \sum_{r=1}^{r_{\max}} \rho_r (1 - x_t)^{r-1}, \tag{II.26}$$

$$x_{t+1} = \epsilon \lambda (1 - y_t). \tag{II.27}$$

Eqn. (II.26) is due to the observation that all the incoming messages at a check node are independent, due to the tree assumption, 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 degree on the left and right it is shown that a graph chosen randomly from the ensemble (N, λ, ρ) is a tree with probability approaching 1 asymptotically in blocklength of the code.

Now if we consider an edge e connected to check node of degree r in the T-peeling 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 is non-erasure, denoted by $y_{t,r}$, if the incoming message on the remaining $r - 1$ edges is an erasure with probability x_t is equal to

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

Averaging over all edges where an edge is connected to a check node of degree r with

probability ρ_r gives us Eqn. (II.17).

II.E.3 Lattice decoding based analysis for compressed sensing

In this appendix we present an alternate analysis for the compressed sensing problem we encountered in Sec II.C.1 based on lattice decoding. A low probability of error for event $\mathcal{E}_{\mathrm{p}}$ for low values of T translates to designing a sensing matrix $\mathbf{A}$ where we require:

1. A large minimum distance in the Euclidean space between distinct T-sums of columns and

2. a minimal number of T-sets of columns whose sum is identical.

Before we formalize the above mentioned notions, we would like to note that, for the choice of $\mathbf{A}$, we considered the superimposed codes proposed by authors Fan, Darnell and Honary for the multiaccess binary adder channel [23]. In this work the authors consider binary codes and show that every constant weight code with weight w and maximum correlation c corresponds to a subclass of disjunctive code of order $T < \frac{w}{c}$. In other words, for any $T < \frac{w}{c}$ sum of any T codewords from this code results in a distinct output. Although the superimposed codes solve the second requirement we mentioned above they do not consider the first requirement i.e., the larger minimum distance of the resulting signal space of T-sums of codewords which is also critical in obtaining a low probability of decoding error values. We present the discussion of these results and our result relaxing the constraint of *constant weight* in Appendix II.E.4.

In the following subsection we introduce lattice and derive upper bounds on $\Pr(\mathcal{E}_{\mathrm{p}})$ based on maximum-likelihood decoder for lattices.

Definition 14. A lattice Λ in n-dimensional Euclidean space $\Lambda \subset \mathbb{R}^n$ can be defined

as:

$$\Lambda = \{\lambda \in \mathbb{R}^n : \lambda = \mathbf{G}\mathbf{u}, \mathbf{u} \in \mathbb{Z}^m\} \tag{II.28}$$

where $\mathbf{G} \in \mathbb{R}^{n \times n}$ is called the generator matrix of the lattice. We define the minimum distance $d_{\min}(\Lambda)$ of the lattice Λ as

$$d_{\min}(\Lambda) \triangleq \min_{\lambda_1, \lambda_2 \in \Lambda} ||\lambda_1 - \lambda_2||_2.$$

Let the set of codewords/columns of $\mathbf{A}$ be denoted by $\mathcal{C}$ and $\mathcal{C} \subseteq \mathcal{C}_{\text{lin}}$ where $\mathcal{C}_{\text{lin}}$ is a binary linear code. We can then observe that the set of T-sums of codewords is a subset of lattice formed from $\mathcal{C}_{\text{lin}}$ ie..,

$$\sum_{j=1}^{T} \vec{a}_{i_j} \in \Lambda \quad i_j \in [1 : M_{\text{p}}]$$

where $\Lambda = \{\mathbf{G}\mathbf{u}, \mathbf{u} \in \mathbb{Z}^m\}$, $\mathbf{G}$ is the generator matrix of the binary code $\mathcal{C}_{\text{lin}}$. Now that the connection between the T-sums of the binary code and the lattice in which they are contained in is established we formalize the two requirements on $\mathbf{A}$ mentioned above.

Definition 15. For a given binary code $\mathcal{C}$ and fixed T, for a subset S of size T, we define the indicator parameter

$$\beta_T(S) \triangleq \mathbf{1}[\exists S' \text{ s.t. } \; u(S) = u(S'), |S'| = T, S' \neq S],$$

where $u(S) := \sum_{i \in S} \vec{c}_i$ and $\beta_T(S)$ indicates if the T-sum of codewords for the index set S is unique in the set of T-sums of codewords from $\mathcal{C}$. The second requirement mentioned above translates to minimizing $\beta_T(\mathcal{C})$ where we define $\beta_T(\mathcal{C}) \triangleq \sum_{S \subset [1:|\mathcal{C}|]} \beta_T(S)$ that counts the total number of subsets whose sum is not unique in

the set of T-sums of codewords from $\mathcal{C}$.

Definition 16. For a given binary code $\mathcal{C}$, fixed T, we define the minimum Euclidean distance of a set S in the space of T-sums of codewords as

$$d_{\min}(S;\mathcal{C}) \triangleq \min_{S\neq S', |S|=|S'|=T} ||u(S) - u(S')||_2 .$$

Also the following relation combining the three quantities above can be observed:

$$d_{\min}(S;\mathcal{C}) \begin{cases} \geq d_{\min}(\Lambda) & \text{if } \beta_T(S) = 0 \\ = 0 & \text{otherwise.} \end{cases} \tag{II.29}$$

We will upper bound the probability of decoding error for the CS problem in terms of the parameters defined in Def. 15 and 16.

Lemma 17. Let $\mathcal{C} \subseteq \mathcal{C}_{\text{lin}}$, where $\mathcal{C}_{\text{lin}}$ is a linear code containing $\mathcal{C}$, be a binary code with parameters $(n, M, d_{\min})$. The probability of error of the bounded distance decoder in decoding $\vec{z} = \sum_{i\in S, |S|=T} \vec{c}_i + \vec{n}$ where $\vec{n} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ can be upper bounded by

$$Pr(\mathcal{E}_{\text{p}}) \leq \frac{\beta_T(\mathcal{C})}{\binom{|\mathcal{C}|}{T}} + \left(\frac{e d^2_{\min}(\Lambda)}{4\sigma^2 N} e^{\frac{-d_{\min}(\Lambda)^2}{4\sigma^2 N}} \right)^{N/2}$$

where N is the blocklength of the code $\mathcal{C}$.

Proof. We recall that the error event $\mathcal{E}_2$ is defined as the event in which the CS decoder fails to decode the set S exactly from

$$\vec{y} = \sum_{i\in S} \vec{a}_i + \vec{z}$$

where $|S| = T$. When we condition event $\mathcal{E}_2$ on the T-sum of vectors from S not being unique, which happens with probability $\frac{\beta_T(\mathcal{C})}{\binom{|\mathcal{C}|}{T}}$ the first part of the bound is obtained. If we assume that the T-sum of vectors from S is unique, then the probability of error in decoding the set S under bounded distance decoding can be upper bounded by $\Pr\left[||\vec{z}|| \geq \frac{d_{\min}(\Lambda)}{2}\right]$ which is equivalent to

$$\Pr\left[\sum_{i=1}^{N} z_i^2 \geq \frac{d_{\min}^2(\Lambda)}{4\sigma^2}\right]$$

where $z_i \sim \mathcal{N}(0,1)$. The result is obtained by using the right tail bounds of Chi-squared distribution. □

We should note that it is not easy to compute the values of $\beta_T(\mathcal{C})$ especially for higher values of T or M_{p}. However sharper conditions for T-disjunctive codes provided in Appendix II.E.4 hopefully provide guidelines to design codes such that $\beta_T(\mathcal{C}) = 0$

II.E.4 T-Disjunctive codes

In the following subsection we first present the main results from [23] that enabled the authors to show that constant weight codes are a subclass of disjunctive code. Then we follow it up with our result where we relax the constant weight constraint on the code to *nearly* constant weight.

Definition 18. The maximum correlation c of a binary code $\mathcal{C}$ is defined as

$$c = \max_{\vec{c}_i, \vec{c}_j \in \mathcal{C},\ i \neq j} < \vec{c}_i, \vec{c}_j > .$$

Definition 19. A binary vector $\vec{c} = [c(1), c(2), \ldots, c(n)]$ is said to be included in a vector $\vec{z} = [z(1), z(2), \ldots, z(n)]$ if and only if $z(i) \geq c(i)\ \forall i$.

Definition 20. A binary code $\mathcal{C}$ with length n, size M is said to be a disjunctive code of order T if each subset $S \subset \mathcal{C}$ with size $|S| \leq T$ has the property that the vector $\vec{z}$ includes only those codewords in $\mathcal{C}$ that belong to S where

$$\vec{z} = \sum_{\vec{c}_i \in S} \vec{c}_i \tag{II.30}$$

is the output of the multiple access real adder channel. We denote a disjunctive code by $D(n, M, T)$.

Definition 21. A constant weight(CW) binary code is one in which all the codewords have equal weight w. For a CW code, the minimum distance $d_{\min}$ and the maximum correlation c are related as

$$2c = 2w - d_{\min}.$$

We denote a constant code by parameters $\mathrm{CW}(n, M, w, c)$ where n, M are blocklength and size of the code respectively.

Lemma 22 ([23] Theorem 1)**.** A constant weight binary code $\mathcal{C}$ with parameters (n, M, w, c) is also a disjunctive code of order (n, M, T) for all T satisfying

$$T < \frac{w}{c}.$$

Example 23. Consider a Reed-Solomon code $RS(n, k, d_{\min}) = RS(7, 3, 5)$. As described in [23] we construct a constant weight code by mapping each symbol in a

codeword from GF(2^3) to a length 8 binary vector of weight one

$$0 \to 10000000$$

$$1 \to 01000000$$

$$\cdots$$

$$7 \to 00000001.$$

Note that this code has parameters $n = 56, M = 2^9, w = 7, d_{\min} = 10$ which implies $c = w - d_{\min}/2 = 2$. Thus any T-sum of the codewords from this CW code is unique for all $T \leq 3 < \frac{w}{c}$.

Now we relax the constant weight constraint in Lemma. 22 and give the corresponding bounds on the disjunctive code parameters.

Lemma 24. For a binary code $\mathcal{C}$ with parameters $(n, M, d_{\min}, w_{\max})$, where $w_{\max}$ is the maximum Hamming weight of all the codewords in the code, the maximum correlation between any two codewords can be given by

$$c \leq w_{\max} - d_{\min}/2.$$

Proof. For any two codewords $\vec{c}_i, \vec{c}_j \in \mathcal{C}$ the relationship between correlation, Hamming distance and sum of Hamming weights can be given by

$$d_H(\vec{c}_i, \vec{c}_j) + 2c(\vec{c}_i, \vec{c}_j) = w_H(\vec{c}_i) + w_H(\vec{c}_j)$$

where d_H and w_H are the Hamming distance and weights respectively. By substituting the lower and upper bounds $d_{\min}$ and $w_{\max}$ for the two parameters gives us the required upper bound on maximum correlation of any two codewords of the binary

code. □

Lemma 25. A binary code $\mathcal{C}$ with parameters $(n, M, d_{\min}, w_{\max})$ is also a disjunctive code of order (n, M, T) for all T satisfying

$$T < \frac{w_{\min}}{w_{\max} - d_{\min}/2}. \tag{II.31}$$

where $w_{\min}$ and $w_{\max}$ respectively are the minimum and maximum Hamming weights of all codewords in the code. Note that the values of $d_{\min}$ and $w_{\min}$ are not necessarily equal for non-linear codes.

Proof. Without loss of generality consider a set $S = \{\vec{c}_1, \vec{c}_2, \ldots, \vec{c}_T\}$ of codewords of size T and let the output of the real adder multiple access channel, given by Eqn. (II.30), be $\vec{z}$. Let us consider codeword $\vec{c}_e \in \mathcal{C} \backslash S$ and look at the event in which $\vec{z}$ does not include $\vec{c}_e$. Let $s_{ie} := \{k : c_i(k) = c_e(k) = 1\}\ \forall i \leq T$ and $s_e = \{k : c_e(k) = 1\}$. Since $\vec{z} = \sum_{i \leq T} \vec{c}_i \implies z(k) \geq 1\ \forall k \in \cup s_{ie}$. Hence the condition that needs to be satisfied for $\vec{z}$ to not include $\vec{c}_e$ is that $\exists k : k \in s_e \backslash \cup\, s_{ie}$ which translates to

$$|\cup\, s_{ie}| < |s_e|. \tag{II.32}$$

The inequality in Eq. (II.32) is satisfied when $\sum_i c(\vec{c}_i, \vec{c}_e) < w_H(c_e)$ which is implied by the condition $T c_{\max} < w_{\min}$ and from Lemma. 24 the required result follows. □

Example 26. Consider a binary BCH code $\mathcal{C}$ with parameters $(n, k, d_{\min}) = (63, 10, 27)$. Let the subset $\mathcal{C}_0 \subset \mathcal{C}$ be obtained via the following decomposition:

$$\mathcal{C} = \mathcal{C}_0 \cup \mathcal{C}_1 \quad \text{such that } c \in \mathcal{C}_0 \iff \bar{c} \in \mathcal{C}_1,$$

where $\bar{c} = \mathbf{1} \oplus c$ is the one's complement of c. For the code $\mathcal{C}_0 \backslash \mathbf{0}$, $\mathbf{0}$ being the all-zero

codeword, the weight and distance parameters are computed to be $(w_{\min}, w_{\max}, d_{\min}) = (27, 36, 27)$ for which the bound in Eqn. (II.31) is $T \leq 1$. But numerically we observe that this code produces unique outputs from the MAC adder channel atleast upto values of $T = 3$. The parameters $\beta_T(\mathcal{C}_0)$ and $d_{\min}(\mathcal{C}_0, T)$ are computed numerically for $T \leq 1$ and are given by :

T	$d_{\min}$	$\beta_T(\mathcal{C})$
1	$\sqrt{27}$	0
2	$\sqrt{27}$	0
3	$\sqrt{27}$	0 .

III. RANDOM MULTIPLE ACCESS

III.A Motivation

Recently, there has been a lot of interest in the design of random access strategies for the uncoordinated massive multiple access problem in view of wireless networks. An interesting connection has been established between codes on graphs and the decoding of multiple users (or, collision resolution in multiple access schemes) [8]. Leveraging this connection, results from coding theory have been used to design and analyze various random access strategies. Particularly, it has been shown that in the limit of the number of users becoming asymptotically large, the throughput efficiency of uncoordinated multiple access can be as high as that of coordinated multiple access [9]. In this chapter, we consider the non-asymptotic regime when the number of users is fixed and finite. By extending the finite-length analysis of low density parity check (LDPC) code ensembles [3] to the multiple access case, we analyze the performance of the random access schemes for finite lengths and validate the analysis with numerical simulations.

III.A.1 System model

In the considered system model there are a total of n users currently active each with one packet of information to transmit to the access point. Similar to Ch. II the transmission period is partitioned into sub-blocks, referred to as slots, thus resulting in a similar slotted structure. Let the total number of slots available per round be m. The random access strategy of each user, independent and uncoordinated from other users, can be described as following. Each user k, $k \in [1 : n]$, populates a random variable D_k distributed according to the probability mass function $L(x)$ or equivalently $\Pr(D_k = i) = L_i$. The respective user then chooses D_k time slots uni-

formly at random, with replacement, from the m available slots. We will refer to this framework of picking check nodes randomly as *uniform-with replacement* framework. We can represent the random strategy via bipartite Tanner graphs similar to a LDPC code where the variable nodes represent the n users and the check nodes represent the m available slots. There exists an edge between variable node i and check node j if and only if user i chose to transmit in slot j. Note that we follow the same convention as used in describing LDPC codes for the degree distribution polynomials:

$$L(x) = \sum_{i=1}^{l_{\max}} L_i x^i \tag{III.1}$$
$$\lambda(x) = \sum_{i=1}^{l_{\max}} \lambda_i x^{i-1},$$

where $L(x)$ and $\lambda(x)$ denote variable node degree distributions, from node and edge perspectives respectively. $R(x)$ and $\rho(x)$ are defined similarly for the check nodes.

For a given $n, L(x)$ probability that a randomly generated graph is not decoded

Notation	Parameter represented
n	Total number of users in the system (variable nodes)
m	Number of time slots per one round of communication (check nodes)
$L(x)$	Variable node degree distribution, node perspective
$R(x)$	Check node degree distribution, node perspective
$\lambda(x)$	Variable node degree distribution, edge perspective
$\rho(x)$	Check node degree distribution, edge perspective
P_B	Prb. that n users are not decoded successfully
P_b	Prb. that a random user is not decoded successfully

Table III.1: Summary of parameters encountered in this chapter along with the notation used is given above.

completely by successive interference cancellation decoder (see peeling decoder [22]) referred to as probability of block error P_B. Similarly the probability that a random user in a session is failed to be decoded by the access point as probability of bit error P_b.

III.B Review

We start with a review of the existing results in the analysis of error performance of finite-length LDPC codes over binary erasure channel (BEC) under peeling decoder. Most of these results are due to Amraoui, Montanari and Urbanke [3].

Consider an LDPC (n, λ, ρ) ensemble which can be defined as the ensemble of bipartite graphs with n variable nodes, m check nodes, and edge connections are formed randomly such that the variable and check node d.d's are $\lambda(x)$, $\rho(x)$ respectively. For more details refer [22]. Luby et al, [24] analyzed the peeling decoder and computed expressions for the evolution of expected number of degree-one check nodes as a function of the size of the residual graph, as the peeling algorithm progresses. More precisely, let $\tilde{R}_1(y)$ denote the fraction of degree-one check nodes (as a fraction of m- number of check nodes in the original graph) present in the residual graph. Here the number of degree-one check nodes in the residual graph is given in parametric form where y is a function of the number of edges peeled off. Note that $t = 0$ corresponds to $y = 1$ and $t \to \infty$ corresponds to $y \to 0$. Then

$$\tilde{R}_1(y) = R'(1)\epsilon\lambda(y)[y - 1 + \rho\left(1 - \epsilon\lambda(y)\right)]. \tag{III.2}$$

For a $(3, 6)$ regular LDPC code, the average number of degree-one check nodes given by (III.2) is plotted in Fig. III.1. Note that the figure is reproduced from [3].

The authors [3] demonstrate that (can also be observed from Fig. III.1) the failure of decoder occurs with high probability in two possible scenarios: The first

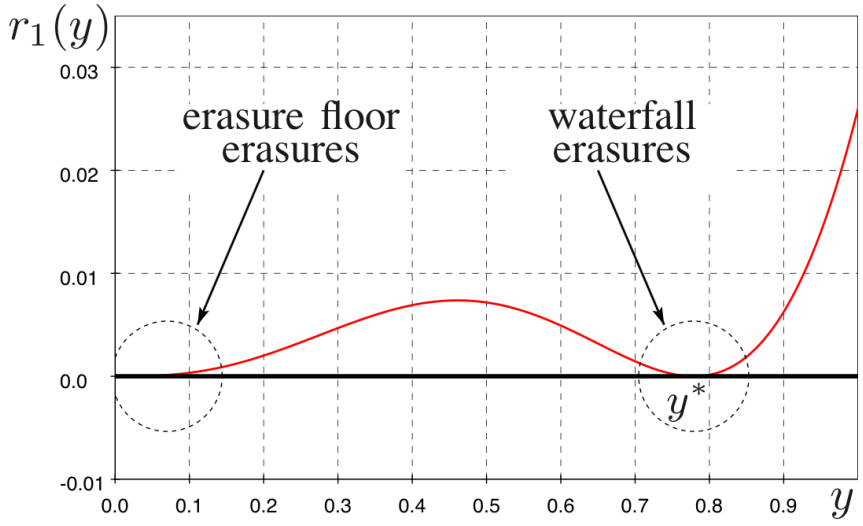

Figure III.1: $\tilde{R}_1(y)$ at $\epsilon = \epsilon_{\text{BP}}$ for $\lambda(x) = x^2, \rho(x) = x^5$. Note that this figure is reproduced from [3] © 2007 IEEE.

case corresponds to $y \approx 0$ or as $t \to \infty$ and the other case corresponds to the value of y such that $\tilde{R}_1(y) = 0$ when $\epsilon = \epsilon_{\text{BP}}$ or equivalently at the value of y where the curve has a stationary point. This point is referred to as critical point y^*. The errors caused corresponding to the first case are referred to as *small-error* events or error floor erasures since they occur towards the end of peeling decoder and the errors corresponding to the second case are referred to as *large-error* events or waterfall erasures. The authors approximate the total probability of error by two expressions, each one corresponding to the one of these two cases.

Lemma 27 (Scaling Law [22])**.** Consider transmission over a BEC channel using random elements from the LDPC (n, λ, ρ) ensemble. Assume that the ensemble has a single critical point $y^* > 0$ and let $\nu^* = \epsilon_{\text{BP}} L(y^*)$. Let $P_{\text{B}}^{W}(n, \lambda, \rho, \epsilon)$ denote the expected block erasure probability due to erasures of size atleast $n\gamma\nu^*$, where

$\gamma \in (0,1)$. Fix $z := \sqrt{n}(\epsilon_{\mathrm{BP}} - \beta n^{-2/3} - \epsilon)$. Then as $n \to \infty$,

$$P_{\mathrm{B}}^{W}(n,\lambda,\rho,\epsilon) = Q\left(\frac{z}{\alpha}\right)\left(1 + O(n^{-1/3})\right)$$
$$P_{\mathrm{b}}^{W}(n,\lambda,\rho,\epsilon) = \nu^* Q\left(\frac{z}{\alpha}\right)\left(1 + O(n^{-1/3})\right)$$

where α and β are constants dependent on the degree distributions.

The expression above approximates the error probability of large-erasure events.

Lemma 28 (Error Floor [3])**.** Consider transmission over a BEC channel using random elements from the LDPC (n,λ,ρ) ensemble. Assume that the ensemble has a single critical point $y^* > 0$ and let $\nu^* = \epsilon_{\mathrm{BP}} L(y^*)$. Let $P_{\mathrm{B}}^{F}(n,\lambda,\rho,\epsilon)$ denote the expected block erasure probability due to stopping sets of size between $s_{\min}$ and $n\gamma\nu^*$, where $\gamma \in (0,1)$. Then for any $\epsilon < \epsilon_{\mathrm{BP}}$

$$P_{\mathrm{B}}^{F}(n,\lambda,\rho,\epsilon) = 1 - e^{-\sum_{s \geq s_{\min}} \tilde{A}_s \epsilon^s}\,(1 + o(1))$$
$$P_{\mathrm{b}}^{F}(n,\lambda,\rho,\epsilon) = \sum_{s \geq s_{\min}} s\tilde{A}_s \epsilon^s\,(1 + o(1))\,,$$

where $\tilde{A}_s = \mathrm{coef}\{\log(A(x)), x^s\}$ for $s \geq 1$, with $A(x) = \sum_{s \geq 0} A_s x^s$ and

$$A_s = \sum_{e} \left(\mathrm{coef}\left\{ \prod_i (1 + xy^i)^{nL_i}, x^s y^e \right\} \times \frac{\mathrm{coef}\{\prod_i \left((1+x)^i - ix\right)^{n(1-r)R_i}, x^e\}}{\binom{nL'(1)}{e}} \right). \quad \text{(III.3)}$$

Note that A_s is the expected number of stopping sets of size s in a random graph chosen uniformly at random from the LDPC $(n\lambda,\rho)$ ensemble and $\tilde{A}_s$ is the expected number of minimal stopping sets of size s. Following along similar lines we derive error floor expression in the case of random multiple access problem.

III.C Error analysis for random multiple access

We note that in the case of random uncoordinated multiple access scheme since the right degree distribution is not a design choice but rather has a Poisson distribution, as discussed in [9], the error probability approximations do not carry over directly from [3] especially for the *water-fall* erasures. But we approximate the small-error events $P_{\mathrm{b}}^{\mathrm{F}}(n, L)$, where 'F' stands for error floor, the expression for which is given in III.C.1. We avoid the large error events by imposing a constraint in the optimization problem that the residual degree-1 check nodes $\tilde{R}_1(y)$ in the initial stages of peeling decoder is bounded away from 0 by a certain threshold and thus the overall probability of error is well approximated by $P_{\mathrm{b}}^{\mathrm{F}}(n, L)$ alone. We support this claim by providing evidence via simulations.

We will refer to the random access framework of picking check nodes randomly but with replacement as *uniform-with replacement* framework. Even though the "uniform-with replacement" allows for multiple edges in the graph, the analysis is made easier because of this assumption. For a given edge, probability that it connects to any of the check nodes is equal to $\frac{1}{m}$. We also believe that the resulting analysis can be easily extended to the "uniform-without replacement" framework where in the D_k check nodes are picked uniformly at random, but without replacement from the m check nodes. Note that in Narayanan, Pfister [9] consider the " uniform-without replacement" framework. Let variable node d.d. $L(x)$, as described in Eqn. (III.1) be the distribution according to which the users choose the repetition degree and the edges are chosen according to the "uniform-without replacement" framework. Under this framework we define the ensemble of graphs for the random multiple access

problem as UMAC(n, λ, η) where $\lambda(x)$ is related to $L(x)$ via

$$L(x) = \frac{\int_0^x \lambda(x)}{\int_0^1 \lambda(x)},$$

η is the throughput ($n = m\eta$).

III.C.1 Error probability approximates

Theorem 29 (Small error events)**.** Consider transmission by users over a noiseless MAC channel according to a graph picked uniformly at random from the UMAC(n, λ, η) ensemble. Assume that the ensemble has single critical point $y^* > 0$ and let $\nu^* = \epsilon_{\mathrm{BP}} L(y^*)$. Let $P^F_{\mathrm{B},s_{\min}}(n, \lambda, \rho, \epsilon)$ ($P^F_{\mathrm{b},s_{\min}}(n, \lambda, \rho, \epsilon)$) denote the expected block (bit) erasure probability due to stopping sets of size between $s_{\min}$ and $\gamma\nu^*$, where $\gamma \in (0, 1)$. Then

$$P^F_{\mathrm{B},s_{\min}}(n, \lambda, \rho, \epsilon) = 1 - e^{-\sum\limits_{s \geq s_{\min}}^{\gamma n} \tilde{A}_s} (1 + o(1)), \tag{III.4}$$

$$P^F_{\mathrm{b}}(n, \lambda, \rho, \epsilon) = \sum_{s \geq s_{\min}} s\tilde{A}_s \epsilon^s (1 + o(1)), \tag{III.5}$$

where $\tilde{A}_s = \mathrm{coef}\{\log(A(x)), x^s\}$ for $s \geq 1$, with $A(x) = \sum_{s \geq 0} A_s x^s$ and

$$A_s = \sum_i \left(\mathrm{coef}\left\{ (1 + x \sum_i L_i y^i)^n, x^s y^i \right\} \times \frac{\mathrm{coef}\{(e^x - x)^m, x^i\}}{\frac{m^i}{i!}} \right). \tag{III.6}$$

Proof. We first show that the expression for A_s in (III.6) is equal to the expected number of stopping sets of size 's' in a graph chosen uniformly at random from the UMAC(n, λ, η) ensemble. The first term is $\binom{n}{s}$ times the probability that s nodes have i edges attached to them and the second term is equal to the probability that

the i edges, under the "uniform-with replacement" framework, form a stopping set i.e, they choose check nodes such that none of the check nodes chosen have only one edge connection.

From there we follow similar argument as in [22] that for large values of n the minimal stopping sets tend to a Poisson distribution with independent components. And then the relation between $A(x)$ and $\tilde{A}(x)$, expression for block/bit error probability follows along the same lines. □

III.C.2 Results

We use the degree distribution given in Eqn. (III.7) with a maximum degree of 30. Note that we chose this distribution randomly. For parameters $n = 1000, m = 1300(\eta \approx 0.77)$ and $L(x)$ in Eqn. (III.7), we obtain the following results given in Table. III.2. Note that $P^{F}_{B,s_{\min}}$ is computed using the analytic expression given in (III.4) whereas $P^{\text{Sim}}_{B,s_{\min}}$ is computed using numeric simulations of peeling decoder.

$$L_1(x) = 0.3x^2 + 0.25x^3 + 0.2x^4 + 0.1x^5 + 0.05x^{10} + 0.04x^{15} + 0.03x^{20} + 0.02x^{25} + 0.01x^{30}. \quad \text{(III.7)}$$

Remark 30. We notice that the analytic and numerical results are almost in perfect agreement. To justify applying Thm. 29 we verify that most of the error events are of small size, analytically through Fig. III.2 by plotting the number of degree-one check nodes in the residual graph. We also verify numerically that the maximum stopping set size we observe is 22 thus rendering an approximation for error probability because of large error events unnecessary.

To verify for another distribution, we perform the experiments for another variable node distribution given in Eqn. (III.8). The evolution of residual degree-one

smin	$P^{F}_{B,\text{smin}}$	$P^{\text{Sim}}_{B,\text{smin}}$	$P^{F}_{b,\text{smin}}$	$P^{\text{Sim}}_{b,\text{smin}}$
2	7.89×10^{-2}	7.97×10^{-2}	2.13×10^{-4}	2.12×10^{-4}
3	2.78×10^{-2}	2.84×10^{-2}	1.05×10^{-4}	1.09×10^{-4}
4	1.11×10^{-2}	1.30×10^{-2}	5.41×10^{-5}	6.33×10^{-5}
5	4.80×10^{-3}	5.90×10^{-3}	2.88×10^{-5}	3.49×10^{-5}
6	2.23×10^{-3}	3.00×10^{-3}	1.58×10^{-5}	2.04×10^{-5}
7	1.10×10^{-3}	1.50×10^{-3}	9.04×10^{-6}	1.14×10^{-5}
8	5.68×10^{-4}	5.00×10^{-3}	5.28×10^{-6}	4.40×10^{-6}
9	3.03×10^{-4}	3.00×10^{-4}	3.15×10^{-6}	2.80×10^{-6}
10	1.66×10^{-4}	1.00×10^{-4}	1.90×10^{-6}	1.00×10^{-6}

Table III.2: Comparison of Probability of Block\Bit errors computed analytically and via simulations for $L_1(x)$ given by Eqn. (III.7), $K = 1000, \eta = 0.77$.

check nodes is given in Fig. III.2. The corresponding numeric results obtained via simulations are given in Table. III.3.

$$L_2(x) = 0.3x^2 + 0.25x^3 + 0.2x^4 + 0.1x^5 + 0.05x^6 + 0.04x^7 + 0.03x^8 + 0.02x^{10} + 0.01x^{20}. \quad \text{(III.8)}$$

smin	$P^{F}_{B,s_{\min}}$	$P^{\text{Sim}}_{B,s_{\min}}$	$P^{F}_{b,\text{smin}}$	$P^{\text{Sim}}_{b,\text{smin}}$
2	7.91×10^{-2}	7.67×10^{-2}	2.13×10^{-4}	2.08×10^{-4}
3	2.79×10^{-2}	2.86×10^{-2}	1.05×10^{-4}	1.12×10^{-4}
4	1.11×10^{-2}	1.28×10^{-2}	5.41×10^{-5}	6.49×10^{-5}
5	4.82×10^{-3}	6.43×10^{-3}	2.88×10^{-5}	3.94×10^{-5}
6	2.25×10^{-3}	3.56×10^{-3}	1.59×10^{-5}	2.51×10^{-5}
7	1.11×10^{-3}	1.63×10^{-3}	9.05×10^{-6}	1.35×10^{-6}
8	5.73×10^{-4}	8.67×10^{-4}	5.29×10^{-6}	8.13×10^{-6}
9	3.06×10^{-4}	5.33×10^{-4}	3.15×10^{-6}	5.47×10^{-6}
10	1.68×10^{-4}	4.00×10^{-4}	1.91×10^{-6}	4.27×10^{-6}

Table III.3: Comparison of Probability of Block\Bit errors computed analytically and via simulations for $L_2(x)$ given by Eqn. (III.8), $K = 1000, \eta = 0.77$.

III.C.3 Necessity of large error approximation

The next obvious question we consider is for what parameters does the probability of small error events given in Eqns. (III.4) & (III.5) is dominated by the large error events and hence is inaccurate estimators of the total probability of error. We consider the case of increases throughput from $\eta = 0.77$ to $\eta = 0.95$ which equates to $m = 1053$ by keeping all other variables in the system same. Before we look at the analytic and numeric results, consider the evolution of residual graph for $\eta = 0.95$ given in Fig. III.2. Notice that for $y = y^* \in (0.82, 0.98)$, the curve is negative implying that with significant probability the error events will be of size concentrating around $\sum_i K\tilde{L}_i(y^*)$, and hence the large error events are non-negligible and in fact will dominate the total error events. To verify this observation numerically we present the results $P^{\text{Sim}}_{B,\text{smin}}$ versus $P^{F}_{B,\text{smin}}$ in Table. III.4.

smin	$P^{F}_{B,\text{smin}}$	$P^{\text{Sim}}_{B,\text{smin}}$	$P^{F}_{b,\text{smin}}$	$P^{\text{Sim}}_{b,\text{smin}}$
2	7.91×10^{-2}	1	2.13×10^{-4}	0.93
3	2.79×10^{-2}	1	1.05×10^{-4}	0.93
4	1.11×10^{-2}	1	5.41×10^{-5}	0.93

Table III.4: Comparison of Probability of Block\Bit errors computed analytically and via simulations for $L_1(x)$ given by Eqn. (III.7), $K = 1000, \eta = 0.95$.

III.D Conclusion

We derived analytic expressions to compute the probability of small error events for the random uncoordinated multiple access problem. We also demonstrated, through numerical simulation, that these analytic expressions are a good estimator for the overall probability of error if the throughput $\eta = \frac{n}{m}$ satisfies certain conditions. The validity of these conditions on throughput can be evaluated by plotting

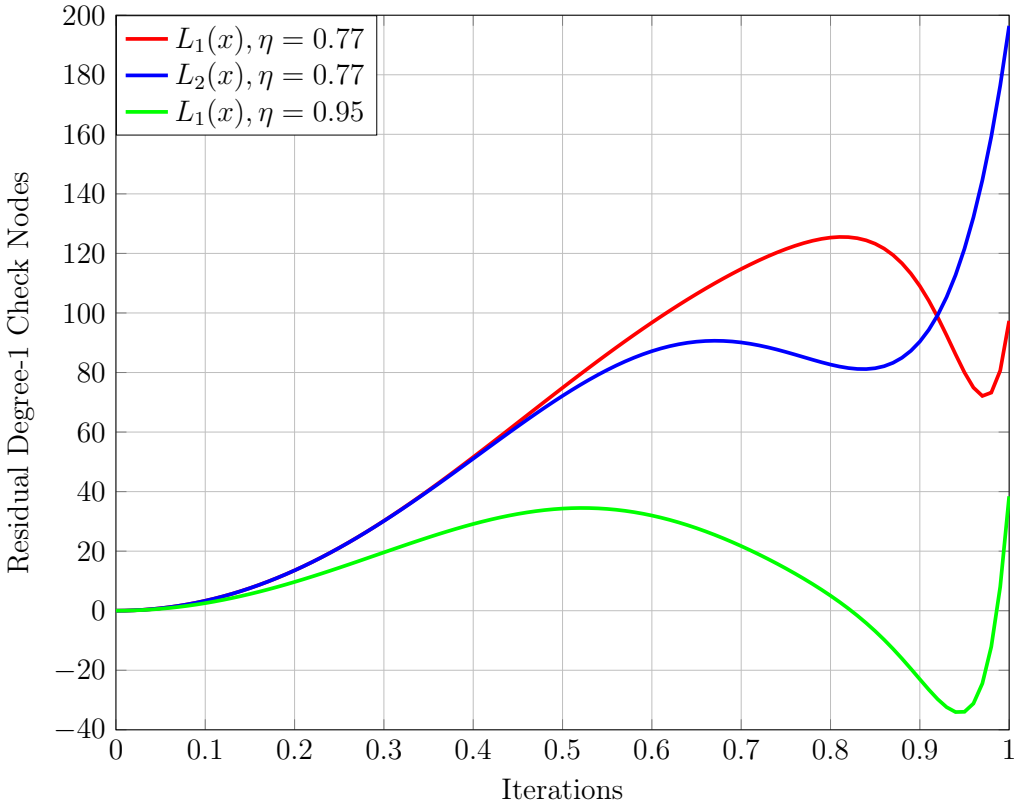

Figure III.2: $\tilde{R}_1(y)$ for UMAC$(1000, L(x), 0.77)$ corresponding to $L_{1,2}(x)$ in Eqn. (III.7).

the evolution of residual degree-one check nodes given by the analytic expression in Eqn. (III.2).

IV. CONSTRUCTION-D LATTICES VIA SPATIALLY COUPLED LDPC CODES*

Lattices have been studied in pure mathematics for more than two centuries. In the last few decades lattices have also found application in connection with coding theory, cryptography and various physical sciences. Codes derived from well designed lattice structures have been shown to be optimal coding solutions to several problems in information and coding theory [25, 26]. In most of these cases, the underlying lattices are constructed using Construction-A and it has been shown that such lattices are simultaneously good for shaping (Roger's good) and for channel coding (Poltyrev good) [25]. There are two important drawbacks in using optimal lattices constructed using Construction-A. On the theoretical side, the use of non-binary codes makes it difficult to prove the optimality of these lattices and lattice codes under practical decoding algorithms such as belief propagation (BP) decoding and so far, we are not aware of any results showing the optimality of Construction-A lattices under BP decoding. On the practical side, optimal lattices constructed from Construction-A typically require the underlying linear codes to work over large fields and hence, result in formidable decoding complexity, even with BP decoding.

In this chapter [27], we discuss Construction-D lattices. We propose a class of lattices constructed using Construction-D [28] where the underlying linear codes are nested binary spatially-coupled low density parity check codes (SC-LDPC) codes with uniform left and right degrees. Forney *et al* [29] showed that the Construction-D lattices achieve the Poltyrev-limit under multi-stage decoding if the underlying

*

codes at each level are capacity achieving. Leveraging this result, and the result due to Kudekar *et al* proving that the regular SC-LDPC codes can universally achieve capacity under BP decoding for the class of binary memoryless symmetric (BMS) channels [30, 31], we show that the proposed Construction-D lattices achieve the Poltyrev limit under multistage BP decoding.

We refer to the proposed lattices as SC-LDPC lattices. The density evolution thresholds show that the proposed SC-LDPC lattices can approach the Poltyrev limit to within 0.2 dB under multistage BP decoding. Around the same time, binary polar codes have been used in conjunction with Construction-D to obtain Poltyrev-good lattices in [32]. The focus of this chapter is on the use of SC-LDPC codes.

We then derive lattice codes from the proposed SC-LDPC lattices and apply them to the symmetric interference channel [33]. It has been pointed out in [34] that there is a natural connection between lattices generated by Construction-D and the interference alignment scheme in [33]. We observe that the interference alignment can be achieved by replacing the Barnes-Wall lattices in [34] by our proposed SC-LDPC lattices.

Throughout the rest of the chapter, vectors and matrices are written in lowercase boldface and uppercase boldface, respectively. $\mathbf{I}_n$ denotes identity matrix of size $n \times n$.

IV.A Lattice preliminaries

A lattice Λ is a discrete set of points in Euclidean space that form an additive group. More precisely an m-dimensional lattice $\Lambda^{(n)} \subset \mathbb{R}^n$ can be defined as:

$$\Lambda^{(n)} = \{\lambda \in \mathbb{R}^n : \lambda = \mathbf{M}\mathbf{u}, \mathbf{u} \in \mathbb{Z}^m\} \tag{IV.1}$$

where $\mathbf{M} \in \mathbb{R}^{n \times m}$ is full-rank and is called the generator matrix of the lattice.

Throughout this chapter, whenever a lattice Λ is used without the superscript, it is understood that the lattice is contained in the n-dimensional Euclidean space i.e., $\Lambda \subset \mathbb{R}^n$. For a given lattice Λ, we denote the *quantizer* with respect to the lattice as Q_Λ, *modulo* operation with respect to the lattice as $\mod \Lambda$, *fundamental Voronoi region* as $\mathcal{V}_\Lambda$ and the *fundamental volume* defined as the volume of any fundamental region as $\mathrm{Vol}(\Lambda)$. For more details on the lattice terminology see [26].

Assume that some $\lambda \in \Lambda$ is transmitted through an additive white Gaussian noise (AWGN) channel of variance σ^2. The *volume-to-noise ratio*(VNR) of Λ, $\alpha^2(\Lambda, \sigma^2)$, is defined as:

$$\alpha^2(\Lambda, \sigma^2) = \frac{\mathrm{Vol}(\Lambda)^{2/n}}{2\pi e \sigma^2}. \tag{IV.2}$$

At the receiver given the decoder $\mathcal{D} : \mathbb{R}^n \to \Lambda$, let us denote the error probability of decoding a lattice point $\lambda \in \Lambda$ as $\mathrm{P}(\lambda, \sigma^2)$ under decoder $\mathcal{D}$. To be more precise,

$$\mathrm{P}(\lambda, \sigma^2) := \Pr\left(\mathcal{D}(\lambda + \mathbf{z}) \neq \lambda\right),$$

where the noise vector is denoted by $\mathbf{z}$. For an infinite lattice Λ, $\mathrm{P}(\lambda, \sigma^2)$ under the minimum Euclidean distance decoder is independent of λ and hence the average probability of decoding error for the lattice $\mathrm{P}(\Lambda, \sigma^2)$ is the same as $\mathrm{P}(\lambda, \sigma^2)$ for any $\lambda \in \Lambda$. Note that minimum distance decoder is the optimal decoder for this problem.

IV.A.1 Poltyrev limit

Definition 31 (Poltyrev Limit)**.** Poltyrev in [35] showed that for any $\delta > 0$ there exists sequence of lattices $\Lambda^{(n)}$, indexed by n, such that the volume-to-noise ratio $\alpha^2(\Lambda^{(n)}, \sigma^2) < 1 + \delta$, $\forall n$ and the average error probability under minimum distance decoder $\mathrm{P}(\Lambda^{(n)}, \sigma^2) \to 0$ as $n \to \infty$. We shall call such a sequence of lattices as being Poltyrev-good.

Remark 32. The converse to the Poltyrev limit i.e., if the VNR of any lattice Λ, $\alpha^2(\Lambda, \sigma^2) < 1$, then it can be shown by simple geometric arguments that the average probability of decoding error is bounded away from 0 even under the minimum distance decoder. Thus the Poltyrev limit is a fundamental limit for the unconstrained AWGN channel coding problem. And also note that to show that a specific sequence of lattices achieve Poltyrev limit it suffices to show that the sequence of lattices satisfy the conditions in Definition 31 using any decoder, not necessarily the optimal minimum distance decoder.

IV.A.2 Construction-D and its goodness

In the literature, even though many lattice constructions such as Construction-A, Construction-D, Construction-D$'$, Construction E etc.., were available since the 1960s, we can safely say that Construction-A has been the most popular one among the information and coding theory communities. The main reason being that the lattices based on Construction-A are used to show (constructive) optimal solutions to various problems like sphere packing, covering, channel coding, source coding, physical-layer network coding etc., [36, 25, 15, 34]. Note that in many of these applications, the optimality is only asymptotic in the field size over which the Construction-A lattice is constructed upon. So even though the Construction-A provides us optimal lattices, the main disadvantage is that at the encoder and decoder should operate over finite fields of very large size. However Construction-D with it's multi-level structure enables us to work over fields of very small size at each level thus making Construction-D lattices much more amenable for practical implementation.

In this subsection we briefly describe multilevel construction of lattices [28, 37], specifically Construction-D and then recall Forney *et al*'s result on the existence of

Poltyrev-good lattices based on this construction [29].

Multilevel construction of lattices is based on a sequence of nested codes $\{\mathcal{C}_l, 1 \leq l \leq r\}$ where each code $\mathcal{C}_l$ is of length n over $\mathbb{F}_q$, a field of size q. Construction-D and Construction-D$'$ are based on such nested sequence of linear codes. For details we refer the reader to [37]. Throughout this work we work with binary linear codes i.e., $q = 2$. For $1 \leq i \leq r$ where r is the number of levels, let $\mathcal{C}_i$ be a (n, k_i) binary code spanned by the set of binary n-tuples $\{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_{k_i}\}$ linearly independent over $\mathbb{Z}$ where $k_1 \leq k_2 \leq \cdots \leq k_r$. One can observe that $\mathcal{C}_1 \subseteq \ldots \subseteq \mathcal{C}_2 \subseteq \mathcal{C}_r$. Using such nested binary linear codes $\{\mathcal{C}_j, 1 \leq j \leq r\}$, a multilevel Construction-D lattice Λ can be defined as follows:

$$\Lambda = \left\{ 2^r\mathbb{Z}^n + \sum_{1\leq i\leq r} 2^{i-1} \sum_{1\leq j\leq k_i} \alpha_{ij}\mathbf{g}_j | \alpha_{ij} \in \{0, 1\} \right\} \tag{IV.3}$$

where "+" denotes addition in $\mathbb{R}^n$. Since the generator vectors $\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_{k_r}$ are all linearly independent and based on the fact that the volume of the Voronoi region corresponding to each point in the lattice is equal to the volume of the fundamental Voronoi region, the VNR of a Construction-D lattice described in (IV.3) can be computed to be

$$\alpha^2(\Lambda, \sigma^2) = \frac{2^{2(r-\sum_{i=1}^{r} k_i/n)}}{2\pi e\sigma^2}. \tag{IV.4}$$

Multistage decoder

We describe the multistage decoding that can be used to decode a Construction-D lattice over any memoryless additive noise channel. Let $\lambda \in \Lambda$, where Λ is as defined in (IV.3), be transmitted through an AWGN channel and $\mathbf{y} = \lambda + \mathbf{z}$ is received where $\mathbf{z} \sim \mathcal{N}(0, \sigma^2\mathbf{I}_n)$. Let $\mathcal{D}_i$ be the component decoder corresponding to $\mathcal{C}_i$ which, given any $\mathbf{x} \in \mathbb{R}^n$, maps it to a codeword in $\mathcal{C}_i$, i.e., $\mathcal{D}_i(\mathbf{x}) \in \mathcal{C}_i$. Let the initialization step

be $\hat{\mathbf{y}}_0 = \mathbf{y}$.

- *Step* 1: At stage i, $1 \leq i \leq r$, $\hat{\mathbf{y}}_{i-1} \mod 2$ is decoded to a codeword $\hat{\mathbf{x}}_i \in \mathcal{C}_i$. Also, the corresponding information bits $\{\hat{\alpha}_{i1}, \hat{\alpha}_{i2}, \ldots, \hat{\alpha}_{ik_i}\} \in \{0,1\}^{k_i}$ that generate $\hat{\mathbf{x}}_i$, are computed.

- *Step* 2: Compute $\hat{\mathbf{y}}_i = \frac{1}{2} \cdot (\hat{\mathbf{y}}_{i-1} - \sum_{1 \leq i \leq k_i} \hat{\alpha}_{ij}\mathbf{g}_j)$. Go to *Step* 1.

- *Step* 3: At $(r+1)^{\text{th}}$ stage of decoding, $\hat{\mathbf{y}}_r$ is decoded to the closest $\mathbf{q} \in \mathbb{Z}^n$ with respect to the Euclidean norm.

- *Output*: Decoded lattice point $\hat{\lambda} \in \Lambda$ is given by

$$\hat{\lambda} = 2^r \mathbf{q} + \sum_{1 \leq i \leq r} 2^{i-1} \sum_{1 \leq j \leq k_i} \hat{\alpha}_{ij}\mathbf{g}_j. \tag{IV.5}$$

At the i^{th} stage of decoding, conditioned on successful decoding in previous stages i.e., assuming $\hat{\alpha}_{pj} = \alpha_{pj}$ for $1 \leq p < i$, the input to the decoder is of the form

$$\begin{aligned}
\hat{\mathbf{y}}_{i-1} \mod 2 &\equiv \frac{1}{2}\Big(\hat{\mathbf{y}}_{i-2} - \sum_{1 \leq j \leq k_{i-1}} \alpha_{(i-1)j}\mathbf{g}_j\Big) \mod 2 \\
&\equiv \Big(\frac{1}{2^{i-1}}\hat{\mathbf{y}}_0 - \sum_{1 \leq p < i} 2^{p-i} \sum_{1 \leq j \leq k_p} \alpha_{pj}\mathbf{g}_j\Big) \mod 2 \\
&\equiv \frac{1}{2^{i-1}}\Big(\mathbf{y} - \sum_{1 \leq p < i} 2^{p-1} \sum_{1 \leq j \leq k_p} \alpha_{pj}\mathbf{g}_j\Big) \mod 2 \\
&\equiv \Big(\sum_{j=1}^{k_i} \alpha_{ij}\mathbf{g}_j\Big) \mod 2 + \Big(2^{-(i-1)}\mathbf{z}\Big) \mod 2 \\
&\equiv \mathbf{x}_i + 2^{-(i-1)}\mathbf{z} \mod 2,
\end{aligned} \tag{IV.6}$$

where $\mathbf{x}_i \in \mathcal{C}_j$. We call the channel defined in (IV.6) as an additive mod-2 Gaussian noise (AMGN) channel [29] and denote the capacity for this channel as $\mathrm{C}_{\mathrm{AMGN}}(\sigma_i^2)$

where $\sigma_i^2 = 2^{-2(i-1)}\sigma^2$.

Theorem 33 (Forney *et al.* [29])**.** For an AWGN channel with noise variance per dimension σ^2, there exists a sequence of Construction-D lattices Λ based on a chain of r two-way one-dimensional lattice partitions and r nested random binary linear codes $\mathcal{C}_1 \subseteq \mathcal{C}_2 \cdots \subseteq \mathcal{C}_r$ that is Poltyrev-good.

Remark 34. Note that in [29], it was shown that if for each level $j \in \{1, \ldots, r\}$ the binary linear code $\mathcal{C}_j$ at that respective level is arbitrarily close to the capacity of the respective AMGN channel and has an arbitrarily low error probability in that stage of the multistage decoder then it was shown that the Construction-D lattice can thus be constructed arbitrarily close to the Poltyrev-limit with an arbitrarily low probability of error.

IV.B Proposed SC-LDPC lattices

As we can see from (IV.3), construction of lattices based on Construction-D using SC-LDPC codes requires the spanning sets, or equivalently generator matrices, of the respective codes to be nested. In other words we need a sequence of SC-LDPC codes where each code is nested in the next code of the sequence. In this section, we first construct such a sequence of nested linear codes where each code has the structure similar to a SC-LDPC system and hence has good error performance at rates arbitrarily close to Shannon capacity. For ease of exposition, we restrict our description to the case $r = 2$. For higher values the construction extends naturally.

IV.B.1 Construction

Let the required rates of the two codes be r_1 and r_2, $0 < r_1 < r_2$. Prior to the details, let us recall that the Tanner graph of a rate $\frac{k}{n}$ binary linear code is the bipartite graph whose $n-k$ check nodes represent the parity check equations defining

the code and the n variable nodes represent the bits of the code. Our objective is to construct Tanner graphs $\mathcal{G}_1, \mathcal{G}_2$ similar in structure to a SC-LDPC system(reference needed?) such that the binary codes $\mathcal{C}_1, \mathcal{C}_2$ represented by the respective graphs are nested i.e., $\mathcal{C}_1 \subseteq \mathcal{C}_2$ and the rates are arbitrarily close to r_1 and r_2 respectively. For small enough $\epsilon > 0$, choose $d_c \in \mathbb{N}$ such that there exists $d_v^1, d_v^2 \in \mathbb{N}$,

$$d_v^i \geq 3 \quad \text{and} \quad 1 - \frac{d_v^i}{d_c} > r_i - \epsilon, i \in \{1, 2\}. \tag{IV.7}$$

The ϵ leeway in the above equations is just for rounding off r_i to the nearest rational number and not very significant.

In this approach we first construct a regular (d_v^1, d_c) Tanner graph $\mathcal{G}_1$ where regularity here means that all the variables have degree d_v^1 and all checks have degree d_c. Then we obtain the Tanner graph $\mathcal{G}_2$ by removing a fraction of the parity checks and the edges incident on these checks in a systematic fashion that $\mathcal{G}_2$ is (d_v^2, d_c) regular. The ensemble described in [38] is not directly amenable to our approach of deriving the higher rate code, since removing a fraction of the checks from this ensemble does not result in a regular SC-LDPC code. Therefore, our approach is to use the following multi edge-type construction.

Fix $M \in \mathbb{N}$. We place Md_c variable nodes at each position in the range $[1 : L] := \{1, 2, \ldots, L\}$, $L \in \mathbb{N}$ and Md_v^1 check nodes at each position in the range $[1 : L+w-1]$, where $w \in \mathbb{N}$ is coupling width. At each position divide the Md_v^1 check nodes into d_v^1 groups where each group contains M check nodes. At any position we refer to all check nodes belonging to k^{th} group as of type $\mathcal{T}_k$. This equates to, at each position, Md_c edges coming from check nodes of type $\mathcal{T}_k$ for all $k \in [1 : d_v^1]$. Similarly, for each variable node, we arbitrarily classify the d_v^1 edges into types, where k^{th} edge is referred to as type $\mathcal{E}_k$ which equates to Md_c edges of each type at any position. For

a fixed $k \in [1 : d_v^1]$, for all $i \in [1 : L]$, each edge of type $\mathcal{E}_k$ at position i is assigned uniformly at random to a type $\mathcal{T}_k$ check node from positions $[i : i+w-1]$. The main idea is that, for each $k \in [1 : d_v^1]$, if we consider the sub-graph containing only the type $\mathcal{T}_k$ check nodes and variable nodes with single edge (type $\mathcal{E}_k$ edges) the above mimics the construction of a $(1, d_c, L, w)$ ensemble [38] on the sub-graph. This results in a Tanner graph in which every variable node has exactly one edge connected to type $\mathcal{T}_k$ check node, for all $k \in [1 : d_v^1]$. We call such a graph, a *check-uniform connected graph*.

More precisely, the ensemble is defined as follows. Choose M such that $\frac{Md_c}{w}$ is a natural number. Fix $k \in [1 : d_v^1]$ and $i \in [1 : L]$. Choose a permutation $\pi_{i,k}^v$ uniformly at random from the set of permutations on Md_c letters. Under arbitrary indexing of the Md_c variable nodes at position i, for $j \in [0 : w-1]$, assign $\mathcal{E}_k$ type edges of $\pi_{i,k}^v \left(j\frac{Md_c}{w} + 1 : (j+1)\frac{Md_c}{w}\right)$ variable nodes at position i to check nodes at position $i+j$. Under this assignment, ignoring the boundary effects, for each check node type at position $i+j$, the number of edges that come from variable nodes at position i is $\frac{Md_c}{w}$, a w^{th} fraction of the total number of connections. From the check nodes perspective, for $k \in [1 : d_v^1]$, at each position, distribute these edges according to a permutation $\pi_{i,k}^c$ chosen uniformly at random from the set of all permutations on Md_c letters. We call the proposed construction as (d_v^1, d_c, L, w) *check-uniform* SC-LDPC (CU-SC-LDPC) ensemble of codes.

Choose a Tanner graph uniformly at random from the above described (d_v^1, d_c, L, w) CU-SC-LDPC ensemble, call it $\mathcal{G}_1$. Observe that, removal of all check nodes of a particular type, say $\mathcal{T}_{d_v^1}$, from $\mathcal{G}_1$ results in a regular $(d_v^1 - 1, d_c)$ Tanner graph. One can see that removal of all check nodes of types $\mathcal{T}_{d_v^2+1}, \mathcal{T}_{d_v^2+2}, \ldots, \mathcal{T}_{d_v^1}$ from $\mathcal{G}_1$ results in a graph from the (d_c, d_v^2, L, w) CU-SC-LDPC ensemble, which let's refer to as $\mathcal{G}_2$. More importantly, all the check-nodes in the derived graph $\mathcal{G}_2$ are also contained in

$\mathcal{G}_1$ and hence any codeword satisfying all the check constraints in $\mathcal{G}_1$ also satisfies all the check constraints in $\mathcal{G}_2$. Thus we can say that the binary code $\mathcal{C}_1$ defined by $\mathcal{G}_1$ is a sub-code of the binary code $\mathcal{C}_2$ defined by $\mathcal{G}_2$. One can obtain a sequence of nested linear codes $\mathcal{C}_1 \subseteq \mathcal{C}_2 \subseteq \ldots \mathcal{C}_r$ by repeatedly performing the above operation. Given $(d_c, d_v^1, \ldots, d_v^r)$, for each code $\mathcal{C}_1$ from the (d_c, d_v^1, L, w) CU-SC-LDPC ensemble, we can obtain a nested sequence of codes $\mathcal{C}_1, \mathcal{C}_2, \ldots, \subseteq \mathcal{C}_r$ where $\mathcal{C}_i \in (d_c, d_v^i, L, w)$ CU-SC-LDPC ensemble. We call the proposed ensemble of nested sequences of codes as $(d_c, d_v^1, \ldots, d_v^r, L, w)$ CU-SC-LDPC ensemble.

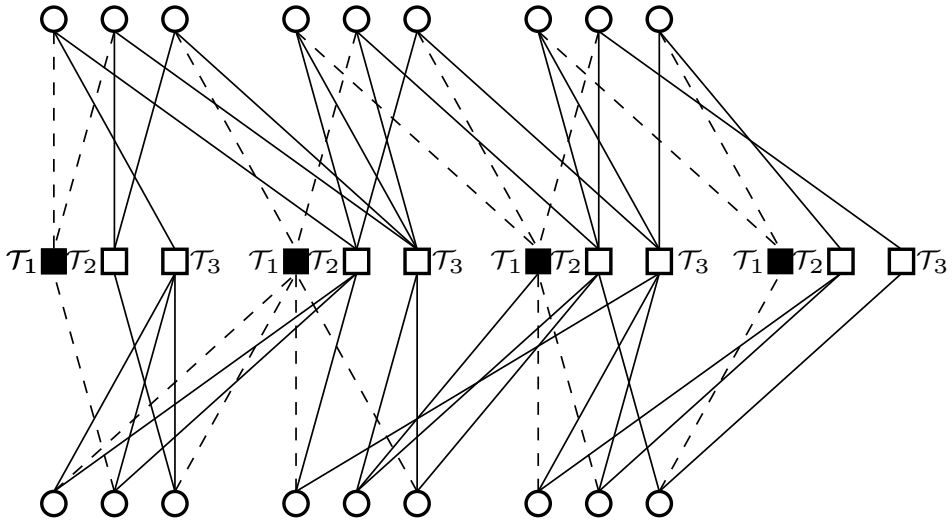

Figure IV.1: A example Tanner graph from the $(3, 6)$, $L = 3$, $w = 2$ CU-SC-LDPC ensemble. Removal of all the type $\mathcal{T}_1$ check nodes i.e the filled nodes, results in a $(2, 6)$ CU-SC-LDPC protograph, see Fig.IV.2.

Remark 35. Observe that choosing $\mathcal{G}_1$ uniformly at random from the (d_c, d_v^1, L, w) CU-SC-LDPC ensemble is equivalent to choosing a set of permutations $\Pi_1 = \{\pi_{i,k}^c, \pi_{i,k}^v : i \in [1 : L], k \in [1 : d_v^1]\}$ where each permutation is chosen uniformly at random from the set of permutations on Md_c letters. Deriving nested graph $\mathcal{G}_2$ is equivalent to

considering just the subset: $\Pi_2 = \{\pi^c_{i,k}, \pi^v_{i,k} : i \in [1:L], k \in [1:d^2_v]\}$ of permutations. Hence this construction of nested codes is equivalent to first constructing $\mathcal{G}_2$ by choosing Π_2 and then choosing $\{\pi^c_{i,k}, \pi^v_{i,k} : i \in [1:L], k \in [d^2_v + 1 : d^1_v]\}$ to construct Π_1 or equivalently $\mathcal{G}_1$.

Example 36. For $r_1 = 0.5$, $r_2 = 0.9$ let us try to compute the degree profiles satisfying (IV.7). One can see that any triplet $(d_c, d^1_v, d^2_v) = (10k, 5k, k), k \geq 3$, satisfies all the required conditions and $(30, 15, 3)$ is the simplest degree profile. We will see later the justification for choosing, $r_2 = 0.9$ in this example (or in general why a nested super-code of high rate, close to 1, is required).

As we have seen in Example. 36, the CU-SC-LDPC construction requires to work with high degree Tanner graphs. Therefore when one attempts multi-stage decoding on lattices based on the proposed nested CU-SC-LDPC ensemble, BP decoding needs to be carried out on a graph that is not very sparse such as the regular $(15, 30)$ Tanner graph in Example 36. For this purpose, to avoid high degree Tanner graphs, we propose the following alternate construction.

IV.B.2 Alternate construction

In contrast to the previous construction where the check node degree remains constant over all the graphs in the sequence of nested codes, in this construction the variable node degree remains constant over all the graphs in the sequence of nested codes. Similar to the previous construction we explain for the case $r = 2$.

Let the the required rates be r_1, r_2 , $0 < r_1 < r_2 < 1$. For small enough $\epsilon > 0$, choose $d_v \geq 3$ such that there exists $d^1_c, d^2_c \in \mathbb{N}$,

$$d^2_c = q d^1_c, q \in \mathbb{N} \quad \text{and} \quad 1 - \frac{d_v}{d^i_c} > r_i - \epsilon, i \in \{1, 2\}. \tag{IV.8}$$

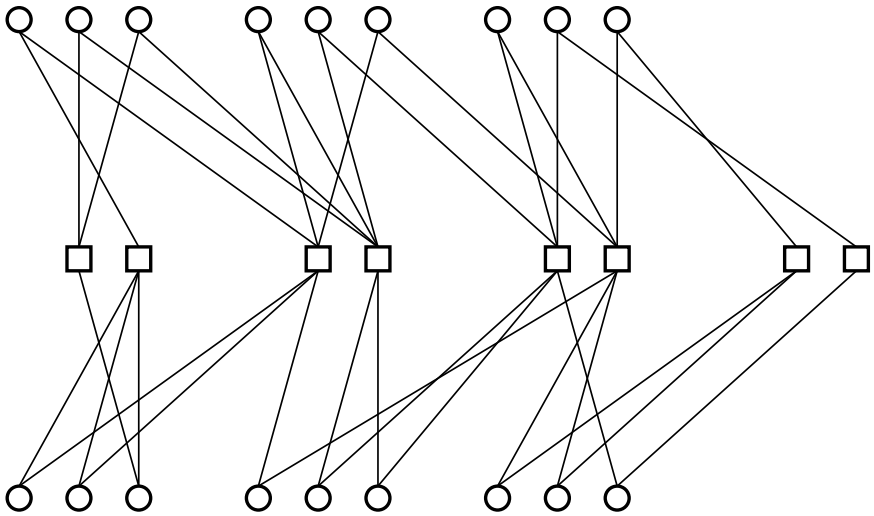

Figure IV.2: A $(2, 6)$ SC-LDPC sub-graph of the $(3, 6)$ SC-LDPC graph shown in Fig.IV.1.

In this construction of a nested pair of $\mathcal{C}_1 \subseteq \mathcal{C}_2$ with SC-LDPC like structure, we first describe the construction of $\mathcal{C}_2$ and then the procedure of deriving the sub-code $\mathcal{C}_1$. The construction of $\mathcal{C}_2$ is identical to that of the (d_v, d_c^2, L, w) SC-LDPC ensemble described in [38]. For sake of being self-contained we briefly describe the construction here. For details we refer the reader to [38].

Fix M such that $\frac{Md_v}{d_c^2} \in \mathbb{N}$. We place M variable nodes at positions $[1 : L]$, $L \in \mathbb{N}$ and $\frac{Md_v}{d_c^2}$ check nodes at positions $[1 : L + w - 1]$, where $w \in \mathbb{N}$ is the coupling width. From the variable node perspective we assign the edges such that each of the d_v connections of a variable node at position i is chosen uniformly at random from the range $[i : i + w - 1]$. Ignoring the boundary effects, the above assignments are such that, for check nodes at position i, the number of edges that come from variable nodes at position $i - j$, $j \in [0 : w - 1]$ is $\frac{Md_v}{w}$. In other words it is exactly w^{th} fraction of the total number of edges at position i. We distribute these edges to $\frac{Md_v}{d_c^2}$ check nodes at position i according to a permutation π_i chosen uniformly at

random from the set of permutations on Md_v letters. With this we can also assume that each of the d_c^2 connections of a check node at position i is independently chosen from the range $[i - w + 1 : i]$. Until this point the construction is identical to the (d_v, d_c^2, L, w) ensemble described in [38], which we hereafter refer to as (d_v, d_c^2, L, w) SC-LDPC ensemble.

We will now describe the construction of a nested sub-code contained in a code from the above ensemble. Let a graph $\mathcal{G}_2$ be picked uniformly at random from the (d_v, d_c^2, L, w) SC-LDPC ensemble and let the binary code defined by this Tanner graph be $\mathcal{C}_2$. Consider a check node C in $\mathcal{G}_2$ and replace the check node C by check nodes $C_1, C_2, \ldots, C_q$ where each new check node C_i has a degree d_c^1(Recall: $d_c^2 = qd_c^1, q \in \mathbb{N}$, by design). With an arbitrary ordering of the d_c^2 edges incident on C, distribute these edges to the new checks $C_1, C_2, \ldots, C_q$ according to a partition Π_C picked uniformly at random from the set of all "partitions of a set of qd_c^2 letters into q subsets of equal size". Note that this operation, which we refer to as *check-splitting*, does not alter the degree of any variable node in the graph. By performing the *check-splitting* operation on all the check nodes in $\mathcal{G}_2$, we derive a regular (d_v, d_c^1) tanner graph. Let the derived graph be denoted $\mathcal{G}_1$ and let the binary code defined by $\mathcal{G}_1$ be $\mathcal{C}_1$.

Lemma 37. $\mathcal{C}_1 \subseteq \mathcal{C}_2$.

Proof. Let c be a check node in $\mathcal{G}_2$ and $\mathbf{h}_c$ be the corresponding parity-check vector (corresponding row in parity-check matrix). As each check node c in $\mathcal{G}_2$ is replaced by check nodes $\{c_1, \ldots, c_q\}$ in $\mathcal{G}_1$ let their corresponding parity check vectors be $\{\mathbf{h}_{c_1}, \ldots, \mathbf{h}_{c_q}\}$. Clearly, from the construction,

$$\mathbf{h}_c = \mathbf{h}_{c_1} + \ldots + \mathbf{h}_{c_q}.$$

Consider $\mathbf{x} \in \mathcal{C}_1$ and a check node $c \in \mathcal{G}_2$. As $\mathbf{x}$ satisfies all the parity-check equations in $\mathcal{G}_1$, $\mathbf{h}_{c_i}^T \cdot \mathbf{x} \equiv 0 \mod 2, 1 \leq i \leq q$.

$$\begin{aligned}\mathbf{h}_c^T \cdot \mathbf{x} &= \left[\mathbf{h}_{c_1}^T + \ldots + \mathbf{h}_{c_q}^T\right] \cdot \mathbf{x} \\ &\equiv 0 \mod 2.\end{aligned}$$

The above implies $\mathbf{h}_c^T \cdot \mathbf{x} \equiv 0 \mod 2, \ \forall c \in \mathcal{G}_2$ and hence $x \in \mathcal{C}_2$. □

One can obtain a sequence of nested linear codes $\mathcal{C}_1 \subseteq \mathcal{C}_2 \subseteq \ldots \subseteq \mathcal{C}_r$ by repeatedly performing the above operation, starting from $\mathcal{C}_r$. We observe that in this construction, unlike the previous construction, for any code $\mathcal{C}_2$ from the (d_v, d_c^2, L, w) SC-LDPC ensemble, choice for deriving a sub-code is not unique. The non-uniqueness arises from the fact that for each check node 'c' the number of choices for the partition Π_c is not unique and any partition will result in a valid sub-code. We call the set of all n-tuples of codes $(\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r)$, where $\mathcal{C}_1 \subseteq \ldots \subseteq \mathcal{C}_r$ derived from the above construction as $(d_v, d_c^1, \ldots, d_c^r)$ VC-SC-LDPC ensemble of nested codes. Here VC refers to *variable-constant* since the variable degree remains constant across the nested sequence of codes. Here after whenever a nested chain of codes is referred to as SC-LDPC and no distinction between the constructions CU-SC-LDPC or VC-SC-LDPC is made, it is implied that the statement is valid for both the constructions.

Remark 38. Note that, given a code $\mathcal{C}_2$ from the (d_v, d_c^2, L, w) SC-LDPC ensemble, deriving a sub-code $\mathcal{C}_1$ is equivalent to choosing a set of partitions: $\{\Pi_c$: 'c' is a check node in $\mathcal{C}_2\}$ uniformly at random. Therefore we can say that in this construction, for any choice of $\mathcal{C}_2$ there are equal number of choices for $\mathcal{C}_1$ from the (d_v, d_c^1, L, w) SC-LDPC ensemble which are all equal likely.

Example 39. Under the VC-SC-LDPC construction, let's consider the same desired

rates $r_1 = 0.5$ and $r_2 = 0.9$ as in Example 36. By simple inspection one can see that the parameters $(d_v, d_c^1, d_c^2) = (3, 6, 30)$ gives us nested VC-SC-LDPC codes of desired rates. Here we need to work with $(3, 6)$ and $(3, 30)$ SC-LDPC codes compared to $(15, 30)$ and $(3, 30)$ SC-LDPC codes in Example 36 based on the CU-SC-LDPC construction.

Since Construction-D works with generator matrices of nested linear codes, we have to obtain nested generator matrices from the proposed nested SC-LDPC codes. In the following lemma, we show the existence of such nested generator matrices for any set of nested binary linear codes.

Lemma 40. Given nested binary linear codes $\mathcal{C}_1 \subseteq \mathcal{C}_2 \subseteq \ldots \subseteq \mathcal{C}_r$ there exists nested generator matrices for these codes.

Proof. It suffices to consider the case having only two levels. For $\mathcal{C}_1$ there exists set of linearly independent binary vectors $\mathbf{G}_1 = \{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_{k_1}\}$ that span $\mathcal{C}_1$ where $k_1 = \dim(\mathcal{C}_1)$. Denote $\mathbf{Z}_i = \{\mathbf{G}_1, \mathbf{g}_{k_1+1}, \mathbf{g}_{k_1+2}, \ldots, \mathbf{g}_{k_1+i-1}\}$ and $Y_i = \mathcal{C}_2 \backslash \text{ span}(\mathbf{Z}_i)$ for $i = 1, 2, \ldots, k_2 - k_1$. Note that for any $\mathbf{x} \in Y_i$, $\mathbf{x}$ is linearly independent with $\mathbf{Z}_i$ and hence $\mathbf{Z}_{i+1} = \{\mathbf{Z}_i, \mathbf{g}_{k_1+i}\}$ forms a linearly independent set where $\mathbf{g}_{k_1+i} = \mathbf{x}$. This recursive procedure gives us a basis $\mathbf{G}_2$ for $\mathcal{C}_2$. Thus the existence of the generator matrices for nested binary linear codes is shown. □

From Lemma 40, given nested SC-LDPC codes $\mathcal{C}_1 \subseteq \mathcal{C}_2 \ldots \subseteq \mathcal{C}_r$, one can find a corresponding sequence of nested sets of generator vectors $\mathbf{G}_1 \subseteq \mathbf{G}_2 \subseteq \ldots \subseteq \mathbf{G}_r$ and hence one can use Construction-D described in (IV.3) with the proposed nested SC-LDPC codes. We refer to the lattice thus constructed as SC-LDPC lattice. Whenever required, we will make the distinction of the lattice being constructed from nested CU-SC-LDPC (or VC-SC-LDPC) sequence of codes by referring it to as a CU-SC-LDPC lattice (or VC-SC-LDPC lattice).

Remark 41. For any code from the (d_v, d_c, L, w) SC-LDPC ensemble, the design rate can be computed to be

$$\mathrm{R}(d_v, d_c, L, w) = 1 - \frac{d_v}{d_c}\frac{L+w-1}{L}. \tag{IV.9}$$

Similarly we define the design VNR of the proposed SC-LDPC lattices to be

$$\alpha_*^2(\Lambda, \sigma^2) = \frac{2^{2(r-\sum_{i=1}^{r} R_i)}}{2\pi e\sigma^2}.$$

where R_i is the design rate of the i^{th} code in the nested sequence of SC-LDPC codes. Although the design rate (IV.9) and the actual rate for any code from the ensemble are not necessarily equal, it is important to observe that the actual rate is atleast as large as the design rate, which gives the following inequality on the actual VNR of the SC-LDPC lattice,

$$\alpha^2(\Lambda, \sigma^2) \leq \alpha_*^2(\Lambda, \sigma^2). \tag{IV.10}$$

IV.B.3 Poltyrev-goodness of the proposed lattices

In this section we show the existence of a sequence of proposed lattices which is Poltyrev-good under BP decoding. In the following lemmas, we show that the proposed SC-LDPC codes (both the constructions) achieve the AMGN channel capacity. We then follow the argument by Forney *et al.* described in Remark 34 to show the result.

Lemma 42. For a BMS channel with associated L-density $\mathbf{x}_{\text{BMS}}$[22], density evolu-

tion (DE) equation for a (d_v, d_c, L, w) CU-SC-LDPC ensemble is given by

$$\mathbf{x}_i^{(l)} = \mathbf{x}_{\mathrm{BMS}} \circledast \left(1 - \frac{1}{w}\sum_{j=0}^{w-1}\left(1 - \frac{1}{w}\sum_{k=0}^{w-1}\mathbf{x}_{i+j-k}^{(l-1)}\right)^{\boxtimes d_c - 1}\right)^{\circledast d_v - 1} \tag{IV.11}$$

where $\mathbf{x}_i^{(l)}$ is the average L-density of the message sent by a variable node at position i in iteration l.

Proof. In the proposed CU-SC-LDPC ensemble, from the perspective of a variable node there are d_v types of edges $\mathcal{E}_1, \mathcal{E}_2, \cdots, \mathcal{E}_{d_v}$. We denote edges of type $\mathcal{E}_k$ that originate from a variable node at position i as $(i, \mathcal{T}_k)$ and the L-density of the message emitted by variable nodes along such edge types as $\mathbf{x}_{ik}^{(l)}$ where l denotes the iteration. But from the perspective of a check node of any type at position i, an edge is randomly connected to one of the variable nodes located at positions $\{i, i-1, ...i-w+1\}$. Hence all the edges connected to check nodes at a certain position are statistically identical and more importantly all check nodes at certain position are statistically identical. The average L-density of the message emitted by a check node at position i in iteration l, denoted by $\mathbf{y}_i^{(l)}$, is given by

$$\mathbf{y}_i^{(l)} = \left(\frac{1}{w}\sum_{j=0}^{w-1}\left(\frac{1}{d_v}\sum_{k=0}^{d_v}\mathbf{x}_{(i-j)k}^{(l-1)}\right)\right)^{\circledast d_c - 1} \tag{IV.12}$$

And a variable node update is given by

$$\begin{aligned}\mathbf{x}_{ik}^{(l)} &= \mathbf{x}_{\mathrm{BMS}} \boxtimes \left(\frac{1}{w}\sum_{j=0}^{w-1}\mathbf{y}_{i+j}^{(l)}\right)^{\boxtimes d_v - 1} \\ \mathbf{x}_i^{(l)} &= \frac{1}{d_v}\sum_{k=0}^{d_v}\mathbf{x}_{ik}^{(l)}\end{aligned} \tag{IV.13}$$

where $\mathbf{x}_i^{(l)}$ is the average L-density of the log-likelihood ratio of variable nodes at

position i. Combining (IV.12) and (IV.13) and observing that the initialization is $\mathbf{x}_i^{(1)} = \mathbf{x}_{i1}^{(1)} = \mathbf{x}_{i2}^{(1)} = \cdots = \mathbf{x}_{id_v}^{(1)} = \mathbf{x}_{\text{BMS}}$ completes the proof. □

Note that in (IV.11) we show that the DE equations for the proposed CU-SC-LDPC ensemble are identical to that of SC-LDPC ensemble proposed in [38, 30].

Remark 43. In the VC-SC-LDPC construction of nested sequence of codes, a (d_v, d_c, L, w) SC-LDPC ensemble is identical to the one in [30] and hence the DE equations in (IV.11) also hold valid for the (d_v, d_c, L, w) SC-LDPC ensemble.

Lemma 44. For any $\delta,\ \epsilon > 0$, there exists parameters d_c, d_v, L, w such that the design rate $\mathrm{R}(d_v, d_c, L, w) > \mathrm{C}_{\text{AMGN}}(\sigma^2) - \delta$ and a code $\mathcal{C}$ from the (d_v, d_c, L, w) CU-SC-LDPC ensemble such that $\mathrm{P_b^{BP}}(\mathcal{C}, \sigma^2) < \epsilon$, where $\mathrm{P_b^{BP}}(\mathcal{C}, \sigma^2)$ is the average bit error probability under BP decoding for $\mathcal{C}$ over AMGN channel with noise variance σ^2 and $\mathrm{C}_{\text{AMGN}}(\sigma^2)$ is the corresponding Shannon capacity.

Proof. It has been proved in [30, 31] that over any BMS channel, under BP decoding, any system that satisfies the equation (IV.11) achieve the capacity as $d_v, w, L \to \infty$ (with $\frac{d_v}{d_c}$ fixed), in that order. Hence if we show that the AMGN channel described in (IV.6) is indeed a BMS channel, then from Lemma 42 it follows that there exist d_c, d_v, L, w large enough such that the design rate $\mathrm{R}(d_v, d_c, L, w) > \mathrm{C}_{\text{AMGN}} - \epsilon$ and the bit error probability $\to 0$ as $M \to \infty$. It is clear to see that the AMGN channel has binary input and output lying in an interval of length 2. Let the input alphabet to the channel be $\{0, 1\}$ and without loss of generality let the $\mod 2$ operation produces a output lying in $[-0.5, 1.5]$. Then the conditional PDFs of $\mathbf{y}$ can be written as

$$f(y|x=0) = \frac{1}{\sqrt{2\pi e\sigma^2}} \sum_{j=-\infty}^{\infty} \exp\left[\frac{-(y+2j)^2}{2\sigma^2}\right] \tag{IV.14}$$

$$f(y|x=1) = \frac{1}{\sqrt{2\pi e\sigma^2}} \sum_{j=-\infty}^{\infty} \exp\left[\frac{-(y+2j-1)^2}{2\sigma^2}\right]. \tag{IV.15}$$

Therefore the PDFs of the output satisfy

$$f(y-0.5|1) = f(0.5-y|0) \quad \text{for all } y \in [-0.5, 1.5].$$

Thus, it belongs to the class of BMS channels. □

Lemma 45. For any $\delta, \epsilon > 0$, there exists d_c, d_v, L, w and code $\mathcal{C}$ from (d_c, d_v, L, w) SC-LDPC ensemble such that the design rate $\mathrm{R}(d_v, d_c, L, w) > \mathrm{C}_{\mathrm{AMGN}}(\sigma^2) - \delta$ and $\mathrm{P}_{\mathrm{b}}^{\mathrm{BP}}(\mathcal{C}_i, \sigma^2) < \epsilon$.

Proof. The proof in Lemma 44 that AMGN channel is BMS and Remark 43 gives us the required result. □

We have shown that there exists good codes from the ensembles of both the constructions, where by a 'good code' we mean a code with the design rate arbitrarily close to the capacity of the AMGN channel and an arbitrarily small probability of error under BP decoding. But for using Construction-D and to be able to apply Forney's result we need to show existence of nested sequence of codes from the proposed constructions where each code in the sequence is a good code. We show this in the following theorems.

Lemma 46. Given r, σ^2, for any $\epsilon > 0$, there exists $d_c, d_v^1, \ldots, d_v^r, L, w$, and a nested sequence of codes $\mathcal{C}_1 \subseteq \mathcal{C}_2 \subseteq \ldots \subseteq \mathcal{C}_r$ from the $(d_c, d_v^1, \ldots, d_v^r, L, w)$ CU-SC-LDPC ensemble such that

$$\mathrm{R}(d_v^i, d_c, L, w) > \mathrm{C}_{\mathrm{AMGN}}\left(\sigma_i^2\right) - 5\epsilon, \text{ and} \tag{IV.16}$$

$$\mathrm{P}_{\mathrm{b}}^{\mathrm{BP}}\left(\mathcal{C}_i, \sigma_i^2\right) < \epsilon \quad \text{for } 1 \leq i \leq r, \tag{IV.17}$$

where $\sigma_i^2 = \frac{\sigma^2}{2^{2(i-1)}}$ is the effective noise variance of the AMGN channel observed at

the i^{th} stage of multi-stage decoding.

Proof. We will prove the result for the case $r = 2$ i.e., the existence of a good nested pair of codes and for the cases $r > 2$ the proof extends naturally. Given $r = 2$, for the given ϵ, δ we choose d_c large enough such that there exists parameters d_v^1, d_v^2, L, w that satisfy (IV.16) simultaneously for $i \in \{1, 2\}$. For these parameters Lemma 44 guarantees us the existence of codes $\mathcal{C}_i \in \mathcal{E}_i := (d_c, d_v^i, L, w)$ CU-SC-LDPC ensemble such that (IV.17) is satisfied for $i \in \{1, 2\}$.

It was not only shown in [30] that any system that satisfies (IV.11) is capacity-achieving asymptotically in d_v, d_c, w, L but also that almost all codes of sufficient length in the ensemble are good over a BMS channel. More precisely, it ([30] Corollary 43) states that for a given $\epsilon > 0$, there exists d_v, d_c, L, w such that $\mathrm{R}(d_v, d_c, L, w) \geq \mathrm{C}_{\mathrm{AMGN}}(\sigma^2) - 5\epsilon$ and

$$\lim_{n\to\infty} \mathbb{E}_{\mathcal{C}(n)\in(d_v,d_c,L,w)} \left[\mathbb{1}_{\{\mathrm{P_b^{BP}}(\mathcal{C}(n),\sigma^2)\leq\epsilon\}} \right] = 1, \tag{IV.18}$$

where the average is over all codes $\mathcal{C}(n)$ of blocklength 'n' from the (d_v, d_c, L, w) SC-LDPC ensemble under uniform distribution. From Lemma 42, (d_v, d_c, L, w) CU-SC-LDPC ensemble satisfies (IV.11) and hence (IV.18) is valid for the (d_v^i, d_c, L, w) CU-SC-LDPC ensembles, $i \in \{1, 2\}$. Hence we can choose n large enough such that

$$\mathbb{E}_{\mathcal{C}_i(n)\in\mathcal{E}_i} \left[\mathbb{1}_{\{\mathrm{P_b^{BP}}(\mathcal{C}_i(n),\sigma^2)\leq\epsilon\}} \right] \geq 1 - \epsilon_1, \tag{IV.19}$$

for $i \in \{1, 2\}$ where the the average is over all codes of blocklength n from $\mathcal{E}_1$.

From Remark 35, this construction is equivalent to first choosing $\mathcal{C}_2$ uniformly at random from $\mathcal{E}_2$ and then choosing $\mathcal{C}_1$ uniformly at random from the set $\mathcal{E}_1(\mathcal{C}_2) := \{\mathcal{C}_1 : \mathcal{C}_1 \in \mathcal{E}_1, \mathcal{C}_1 \subseteq \mathcal{C}_2\}$. From the fact that the set $\mathcal{E}_1(\mathcal{C}_2)$ has same cardinality for all

choices of $\mathcal{C}_2$ (see remark 35), we can deduce that the marginal distribution for $\mathcal{C}_1$ is uniform on $\mathcal{E}_1(\mathcal{C}_2)$.

For being concise we refer to code $\mathcal{C}_i \in \mathcal{E}_i$ as good if $\mathrm{P}_{\mathrm{b}}^{\mathrm{BP}}(\mathcal{C}_i, \sigma^2) \leq \epsilon_2$ and bad if otherwise. Consider the probability of choosing a bad code for either levels,

$$\begin{aligned}
&\Pr[\ \mathcal{C}_2 \text{ is bad or } \mathcal{C}_1 \text{ is bad }] \\
&\quad = \Pr[\ \mathcal{C}_2 \text{ is bad }] + \Pr[\ \mathcal{C}_1 \text{ is bad} | \mathcal{C}_1 \in \mathcal{E}_1(\mathcal{C}_2)] \\
&\quad \leq \Pr[\mathcal{C}_2 \text{ is bad}] + \Pr[\mathcal{C}_1 \text{ is bad} | \mathcal{C}_1 \in \mathcal{E}_1] \\
&\quad \leq 2\epsilon_1
\end{aligned}$$

where the last inequality follows fromEq. (IV.19). This not only gives us the existence of a good pair of nested codes arbitrarily close to capacity but also that almost all nested pairs from (d_c, d_v^1, d_v^2) CU-SC-LDPC ensemble are good. □

Lemma 47. Given r, σ^2, for any $\epsilon > 0$, there exists $d_c, d_v^1, \ldots, d_v^r, L, w$, and a nested sequence of codes $\mathcal{C}_1 \subseteq \mathcal{C}_2 \ldots \subseteq \mathcal{C}_r$ from the $(d_v, d_c^1, \ldots, d_c^r, L, w)$ VC-SC-LDPC ensemble such that (IV.16) and (IV.17) are satisfied.

Proof. In the proof of Lemma 46, using remark 43, Lemma 45 and remark 38 instead of Lemma 42, Lemma 44 and remark 35 respectively gives us the required proof. □

Theorem 48. For any $\epsilon, \delta > 0$, there exists a CU-SC-LDPC lattice Λ with $\alpha^2(\Lambda, \sigma^2) < 1 + \epsilon$ for which, under multistage BP decoding, the average probability of error $\mathrm{P}(\Lambda, \sigma^2) < \delta$.

Proof. We first choose r large enough such that

$$\mathrm{P}\left(\mathbb{Z}_{2^r}^n, \sigma_{r+1}\right) < \frac{\delta}{r+1},$$

where $\mathrm{P}\left(\mathbb{Z}_{2^r}^n, \sigma_{r+1}\right)$ is the average error probability in decoding a point chosen uniformly at random from $\mathbb{Z}_{2^r}^n$ under minimum distance decoder. Then from Lemma 46, there exists $(d_c, d_v^1, \ldots, d_v^r)$ and nested sequence of codes $(\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_r)$ from the $(d_c, d_v^1, \ldots, d_v^r)$ CU-SC-LDPC ensemble such that

$$\mathrm{R}(d_v^i, d_c, L, w) > \mathrm{C}_{\mathrm{AMGN}}\left(\sigma_i\right) - \epsilon_1, \text{ and} \tag{IV.20}$$

$$\mathrm{P}_{\mathrm{b}}^{\mathrm{BP}}\left(\mathcal{C}_i, \sigma_i\right) < \epsilon_2 \qquad \text{for } 1 \leq i \leq r. \tag{IV.21}$$

By union bound,

$$\mathrm{P}^{\mathrm{BP}}(\mathcal{C}_i, \sigma_i^2) < n\ \mathrm{P}_{\mathrm{b}}^{\mathrm{BP}}(\mathcal{C}_i, \sigma_i^2) \tag{IV.22}$$

where $\mathrm{P}^{\mathrm{BP}}(\mathcal{C}_i, \sigma_i^2)$ is the corresponding block error probability. We then choose $\epsilon_2 = \frac{\delta}{n(r+1)}$, use the union bound to bound the total error probability in decoding a lattice point which results in $\mathrm{P}(\Lambda, \sigma^2) < \delta$. Recalling remark 34 and then the Eqn (IV.10) bounding the actual VNR completes the proof. □

Theorem 49. For any $\epsilon, \delta > 0$, there exists a VC-SC-LDPC lattice Λ with $\alpha^2(\Lambda, \sigma^2) < 1 + \epsilon$ for which, under multistage BP decoding, the average probability of error $\mathrm{P}(\Lambda, \sigma^2) < \delta$.

Proof. In the proof of Theorem 48, using Lemma 47 instead of Lemma 46 completes the proof. □

Remark 50. Although lattices based on both the constructions have shown to be Poltyrev-good, both have their own pros and cons (advantages and disadvantages?). The parameters $(d_c, d_v^1, \ldots, d_v^r)$ in the CU-SC-LDPC construction admit any set of natural numbers and thus give greater flexibility in constructing codes of desired rates at each level and thus provides the ability to match the capacity of the effective AMGN channel with a greater accuracy. But on the flip side this results in higher

degree profiles, as explained in example 36, and hence making the decoding more complex. Whereas in the case of VC-SC-LDPC construction, the parameters only admit sets of the form $(d_v, q_1 d_c, q_2 d_c, \ldots, q_r d_c)$, $q_i \in \mathbb{N}$, which is not very flexible when it comes to matching a given rate-tuple. But as we seen in examples 36 and 39, in certain specific cases of desired rates, VC-SC-LDPC offers nested sequence of codes of considerably low complex degree profiles compared to the CU-SC-LDPC construction.

Remark 51 (Comparison with LDPC lattices)**.** LDPC codes have been adopted as underlying codes for constructing lattices in [39] where the so-called LDPC lattices have been proposed and analyzed. Our SC-LDPC lattices differ from LDPC lattices in the following ways. Firstly, LDPC lattices are constructed based on Construction-D$'$ [28] in contrast to Construction-D adopted here. Secondly, our decoding algorithm is a multistage BP decoding which only works over $\mathbb{F}_2$, on the contrary, since constructed based on Construction-D$'$, LDPC lattices have to consider BP algorithm on the joint Tanner graph [40] (i.e., joint decoding). Last but not least, since there are no analytical evidence that LDPC codes under BP decoding would achieve capacity, LDPC lattices have not been shown Poltyrev-good to the best of our knowledge while for the proposed SC-LDPC lattices, Theorems 48 and 49 serves as constructive evidence.

IV.B.4 Design and simulation results

In this subsection, we explain the design of SC-LDPC lattices that approach the Poltyrev limit with examples. Before the design principles let us analyze the decoding error probability.

Let the number of levels required be $r + 1$, with r coded levels using nested SC-LDPC codes and the last level being uncoded using the $\mathbb{Z}^n_{2^r}$ lattice. The design

criteria depend mainly on the target error probability and the dimension of the lattice. For illustration let the target block error probability be $\approx 10^{-4}$ and the number of dimensions be $n = 2 \times 10^5$. As we use multistage decoding the average probability of decoding error $\mathrm{P}(\Lambda, \sigma^2)$ can be union bounded by the sum of block error probabilities at individual levels. Assuming the constituent SC-LDPC code at each level is operating below the BP threshold[22], the average probability of decoding error of the lattice is dominated by the performance of the last (uncoded) level since the class of LDPC codes have a very sharply decaying error probability profiles below the BP threshold. Let's recall that $\mathrm{P}(\mathbb{Z}_{2^r}^n, \sigma^2)$ is the block error probability for the last level. Similar to (IV.22), using union bound,

$$\mathrm{P}(\mathbb{Z}_{2^r}^n, \sigma_{\mathrm{r+1}}^2) \leq n\mathrm{P}(\mathbb{Z}_{2^r}, \sigma^2) = n\left(2Q\left(\frac{0.5}{\sigma_{r+1}}\right)\right). \tag{IV.23}$$

Plugging in the values of n and the target error probability in (IV.23) gives us $\sigma_{r+1} = 0.0804$. Now moving to the next level i.e., level r, $\sigma_r = 2\sigma_{r+1} = 0.1608$. The capacity of the effective AMGN channel observed in this level of the multi-stage decoding is $\mathrm{C}_{\mathrm{AMGN}}(\sigma_r^2) = 0.9923$, see Fig. IV.3. For the details on computing the capacity of the AMGN channel see [29]. Similarly proceeding, $\sigma_{r-1} = 2\sigma_r = 0.3217$, $\mathrm{C}_{\mathrm{AMGN}}(\sigma_{r-1}^2) = 0.5726$, $\sigma_{r-2} = 2\sigma_{r-1} = 0.6434$, $\mathrm{C}_{\mathrm{AMGN}}(\sigma_{r-2}^2) = 0.0242$. Observe that the capacity for level $r-2$ is almost zero which renders coding for this level unnecessary albeit at the cost of a very small increase in VNR (due to the rate loss) of 0.145dB ($= 20\log_{10} 2^{0.0242}$). Hence $r = 2$ i.e., two coded levels suffice. We use $(30, 14, 3)$ CU-SC-LDPC ensemble with $L = 32, w = 4$ for the first two levels and $\mathbb{Z}_4^n$ lattice for the last level which results in nested SC-LDPC codes of rates 0.5333 and 0.9 (0.49 and 0.89 including rate-loss due to boundary effects of coupling) matching closely the capacities of first two levels i.e. 0.5726 and 0.99.

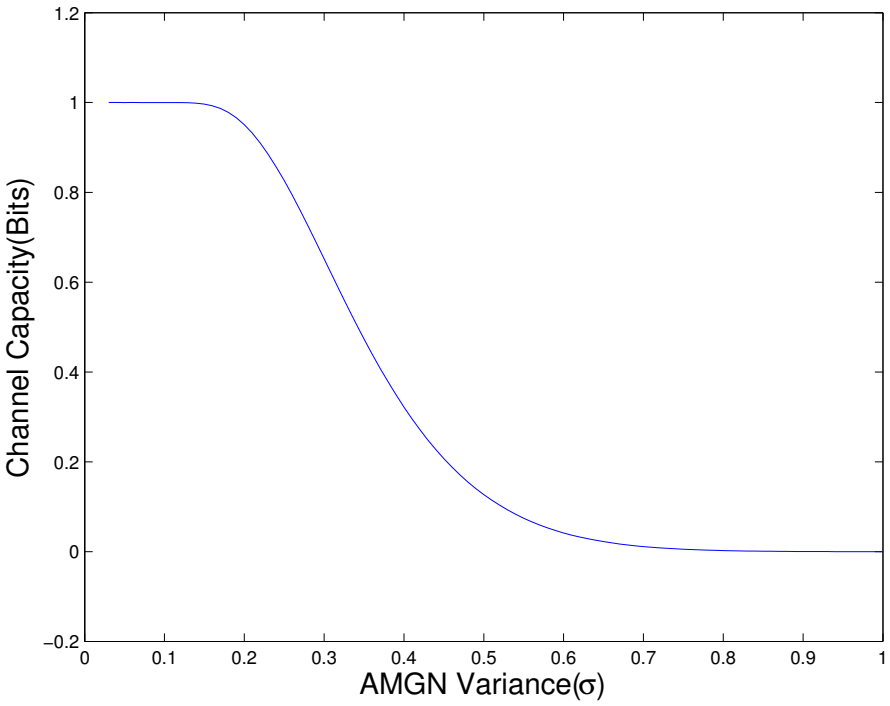

Figure IV.3: Channel capacity of the additive mod-2 Gaussian noise channel

Simulation results

Due to symmetry in the lattice the all-zero lattice point is assumed to be transmitted. Instead of plotting the symbol error rate, we focus on determining the thresholds of the resulting lattice under BP decoding. We estimate the BP threshold from simulations by determining the maximum noise variance for which no codeword errors are observed, at each coded level, in simulation of 10 consecutive codewords each of length 2×10^5. We calculate the maximum variance $\sigma^2_{\max}$ for which all the levels of the lattice can be decoded given by $\sigma_{\max} = \min(\sigma_1^{\mathrm{BP}}, 2\sigma_2^{\mathrm{BP}}, \sigma_3)$, where σ_1^{BP} and σ_2^{BP} are the respective BP thresholds for the two SC-LDPC codes and σ_3^2 is the noise variance at which the uncoded level achieves the target error probability. The

(d_c, d_v^1, d_v^2)	(L,w)	$\sigma_{\max}$	VNR*(dB)	VNR$_{\text{rate-loss}}$(dB)
(30,14,3)	(32,4)	0.3184	1.14	1.347
(60, 27, 3)	(64, 9)	0.3203	0.57	0.951
(60, 26, 3)	(72, 12)	0.3200	0.482	0.927
(60, 42, 3)	(72, 12)	0.3975	0.203	1.02

Table IV.1: Density evolution (DE) thresholds for SC-LDPC lattice ensembles under BP decoding for various degree profiles. The gap from the respective Poltyrev limits, computed without considering rate loss from termination, are also given.

VNR threshold is then calculated for the given rates and $\sigma_{\max}$. Thus obtained BP thresholds σ_1^{BP}, σ_2^{BP} for the above codes are 0.3142 and 0.2161 respectively which results in a VNR of 1.14dB (1.46dB with rate loss due to termination). The DE predicted values are 0.3184 and 0.21836. We observe that the BP thresholds are very close to DE thresholds. i.e., the parameters are large enough to assume that the BP thresholds can be approximated by DE thresholds. Therefore it is reasonable to calculate the VNR thresholds using the DE thresholds. For various SC-LDPC ensembles Table. IV.1 gives us the VNR thresholds i.e., the VNRs achievable for respective target error probabilities which are computed using the DE thresholds. Note that the Poltyrev limit is zero dB, thus making the VNR threshold and the gap from Poltyrev limit equivalent. The gap to the Poltyrev limit is primarily due to the fact that there is a mismatch between the capacity of the equivalent channel and the rates that are obtainable for the proposed CU-SC-LDPC ensemble.

In the above design, if we target a error probability per dimension of 10^{-6} instead, that gives us $\sigma_{r+1} = 0.0999$, capacities for the subsequent levels $\text{C}_{\text{AMGN}}(\sigma_r^2) = 0.9507$, $\text{C}_{\text{AMGN}}(\sigma_{r-1}^2) = 0.3223$ and $\text{C}_{\text{AMGN}}(\sigma_{r-2}^2) = 0.0024$. Pair of nested codes from $(60, 42, 3)$ CU-SC-LDPC ensemble gives us rates 0.3 and 0.95 resulting in better matching of the rates (negligible rate loss). The resulting DE thresholds are within 0.203dB from Poltyrev limit. This is reported in the last row in the table. [Couple

of lines - Broadly justifying the VC-SC-LDPC construction] Observe that for these parameters we can use a $(60, 4, 3)$ VC-SC-LDPC ensemble with the same parameters of $(L = 72, w = 12)$ gives us codes of rates 0.25 and 0.95. Although this results in a slight VNR-loss due to the relatively poor mismatching of rates, in this case BP decoding is carried out on a $(3, 4)$ Tanner graph instead of a $(42, 60)$ which considerably reduces the complexity of decoding.

IV.C Application: Interference channel

IV.C.1 Problem statement

We consider the 3 user Gaussian interference channel (IC) consisting of 3 transmitters, 3 receivers, and 3 independent messages originally considered in [15], where message W_j originates at transmitter j and is intended for receiver j, $\forall j \in \mathcal{J} \triangleq \{1, 2, 3\}$. The output observed at the receiver j is given by

$$\mathbf{y}_j = \mathbf{x}_j + \sum_{k=1, k\neq j}^{3} h_{jk}\mathbf{x}_k + \mathbf{z}_j, \quad \forall j \in \mathcal{J} \tag{IV.24}$$

where $\mathbf{x}_j$ is the transmitted signal at j^{th} transmitter, h_{jk} are the channel parameters for the cross links, and $\mathbf{z}_j \sim \mathcal{N}(\mathbf{0}, \sigma^2 \cdot \mathbf{I})$ is the AWGN noise. If the channel parameters for all the cross links are equal we refer to such model as symmetric IC. The channel input signals are subjected to the power constraint $\frac{1}{n}\sum_{i=1}^{n} E\left[\|\mathbf{x}_j\|^2\right] \leq P$.

For a 2-user symmetric Gaussian interference channel (IC) it was shown in [12] that, in the very strong interference regime, the capacity region for the IC is as if there is no interference at all. For this symmetric model, a simple extension of the very strong interference condition for the 2 user IC to the 3 user one is given by [15]

$$\beta^2 \geq \frac{\left((1+P)^2 - 1\right)(1+P)}{2P}. \tag{IV.25}$$

Sridharan *et al.* in [15] introduced the idea of lattice alignment where each user uses a lattice code and each receiver first decodes the total interference (aligned due to lattice structure) observed and then decodes the desired message. For this case, they derived a tighter condition on β in order for the interference to be decoded first. This is based on lattice coding, independent of the number of users, and is given by

$$\beta^2(\sigma) \geq \beta^{*^2}(\sigma) \triangleq \frac{(P+\sigma^2)^2}{P\sigma^2} \tag{IV.26}$$

If (IV.26) is satisfied, each user can achieve a capacity of $\frac{1}{2}\log(1+\frac{P}{\sigma^2})$ [15]. Equivalently, for a given rate R, maximum noise variance under which the rate can be achieved is given by

$$\sigma^2_{\max} = \frac{P}{2^{2R}-1}. \tag{IV.27}$$

IV.C.2 Applying the proposed lattices

Encouraged by the Poltyrev-limit achieving property of the proposed lattice ensembles under BP decoding, we use SC-LDPC lattice codes for the symmetric Gaussian IC in the very strong interference region. Let Λ_{SC} be the SC-LDPC lattice defined in (IV.3) with $r = 2$. We define the SC-LDPC lattice code $\mathcal{C}_{SCL}$ based on Λ_{SC} using hypercube shaping:

$$\mathcal{C}_{SCL} = \{\lambda \mod \mathbb{Z}_4^n : \lambda \in \Lambda\} \tag{IV.28}$$

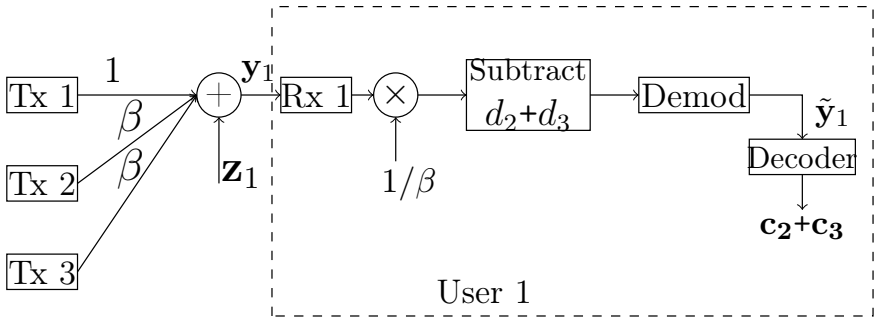

Figure IV.4: System flow for the 3-user Symmetric Gaussian Interference channel at receiver 1.

where n is the dimension of Λ_{SC}. Let codeword $\mathbf{c}_j \in \mathcal{C}_{SCL}$ at transmitter j be

$$\mathbf{c}_j = \sum_{i=1}^{k_1} \alpha_{ji}\mathbf{g}_i + 2\sum_{i=1}^{k_2} \beta_{ji}\mathbf{g}_i \mod \mathbb{Z}_4^n \quad \alpha_{ji}, \beta_{ji} \in \{0,1\} \tag{IV.29}$$

$$= \sum_{i=1}^{k_1} \alpha_{ji}\mathbf{g}_i + 2\sum_{i=1}^{k_2} \beta_{ji}\mathbf{g}_i - 4\mathbf{k}_j, \text{ for some } \mathbf{k}_j \in \mathbb{Z}^n \tag{IV.30}$$

where "+" denotes addition in $\mathbb{R}^n$. Each codeword $\mathbf{c}_j \in \mathcal{C}_{SCL} \subset \{0,1,2,3\}^n$ is modulated to $\tilde{\mathbf{x}}_j \triangleq 1.5^n - \mathbf{c}_j$ such that $\tilde{\mathbf{x}}_j \in \mathcal{A} \triangleq \{-1.5, -0.5, +0.5, +1.5\}^n$. At transmitter j, a dither vector $\mathbf{d}_j$ uniformly distributed among $\mathcal{B} \triangleq [-2, 2)$ is added to obtain the transmitted signal $\mathbf{x}_j$ given by

$$\mathbf{x}_j = \tilde{\mathbf{x}}_j + \mathbf{d}_j \mod \mathbb{Z}_4^n, \tag{IV.31}$$

where the mod operation is over $\mathcal{B}$ instead of $[0, 4)$. The dither vector achieves the purpose of randomizing the interference and helps in treating the undesired components of the received signal as additive uncorrelated noise. It can be seen that $\mathbf{x}_j$ is uniformly distributed over $\mathcal{B}$ and the average power of the transmitted signal at each transmitter is 1.33.

IV.C.3 Decoding

Before looking at the general case let us consider the symmetric Gaussian IC i.e $h_{12} = h_{13}$. Without loss of generality let us consider receiver 1. The system schematic from the perspective of receiver 1 is given in Fig. IV.4. The input to the multistage decoder at receiver 1 is given by

$$\begin{aligned}\tilde{\mathbf{y}}_1 &\triangleq \frac{\mathbf{y}_1}{h_{12}} - \mathbf{d}_2 - \mathbf{d}_3 + 1.5^n + 1.5^n \\ &= \mathbf{c}_2 + \mathbf{c}_3 + \frac{1}{h_{12}}\left(\mathbf{x}_1 + \mathbf{z}_1\right).\end{aligned}$$

Note that $\mathbf{c}_2, \mathbf{c}_3 \in \mathcal{C}_{SCL} \subset \Lambda$ and hence $\mathbf{c}_2 + \mathbf{c}_3 \in \Lambda$.

$$\begin{aligned}\mathbf{c}_2 + \mathbf{c}_3 &= \sum_{i=1}^{k_1} \left(\alpha_{2i} + \alpha_{3i}\right) \mathbf{g}_i + 2\sum_{i=1}^{k_2} \left(\beta_{2i} + \beta_{3i}\right) \mathbf{g}_i + 4\mathbf{k}_2 + 4\mathbf{k}_3 \\ &= \sum_{i=1}^{k_1} \left(\alpha_{2i} \oplus \alpha_{3i}\right) \mathbf{g}_i + 2\sum_{i=1}^{k_2} \left(c_{1i} \oplus \beta_{2i} \oplus \beta_{3i}\right) \mathbf{g}_i + 4\mathbf{k}_{23}\end{aligned}$$

where $c_{1i} = 0.5\left(\alpha_{2i} + \alpha_{3i} - \alpha_{2i} \oplus \alpha_{3i}\right)$, $c_{2i} = 0.5\left(c_{1i} + \beta_{2i} + \beta_{3i} - c_{1i} \oplus \beta_{2i} \oplus \beta_{3i}\right)$ are carryovers from first and second levels respectively and $\mathbf{k}_{23} = \mathbf{k}_2 + \mathbf{k}_3 + \sum_1^{k_2} c_{2i}\mathbf{g}_i \in \mathbb{Z}^n$. The key here is that $c_{1i}, c_{2i} \in \{0, 1\}$ which lets us apply multi-stage BP decoding. Using multi-stage decoder described in Section IV.B, one can directly decode the lattice point $\mathbf{x}_2 + \mathbf{x}_3$(interference), subtract it and decode the desired signal.

The decoding scheme above extends to the case when one channel gain is an integer multiple of the other. For example, let $h_{13} = Kh_{12}$ where $K = \sum_0^{l-1} a_i 2^i \in$

$\mathbb{Z}, a_i \in \{0,1\}$. In this case, input to the multi-stage decoder is

$$\begin{aligned}\tilde{\mathbf{y}}_1 &\triangleq \frac{\mathbf{y}_1}{h_{12}} - \mathbf{d}_2 - K\mathbf{d}_3 + 1.5^n + K1.5^n \\ &= \mathbf{c}_2 + K\mathbf{c}_3 + \frac{1}{h_{12}}\left(\mathbf{x}_1 + \mathbf{z}_1\right).\end{aligned}$$

where $\mathbf{c}_2 + K\mathbf{c}_3$ is a lattice point and is given by

$$\sum_{i=1}^{k_1} \left(\alpha_{2i} \oplus a_0\alpha_{3i}\right)\mathbf{g}_i + 2\sum_{i=1}^{k_2}\left(c_{1i} \oplus \beta_{2i} \oplus a_0\beta_{3i} \oplus a_1\alpha_3\right)\mathbf{g}_i + 4\mathbf{k}$$

for some $\mathbf{k} \in \mathbb{Z}^n$.

IV.C.4 Simulation results for symmetric IC

In this section we present simulation results for the symmetric Gaussian IC and compare them with the bounds given in [15]. We choose a pair of nested codes from the $(30, 18, 3)$ CU-SC-LDPC ensemble with spatial-coupling parameters $(L, w) = (32, 4)$. We fix $\sigma = \sigma_{\max}$ (such that in absence of interference, desired signal can be decoded successfully) and we analyze the bit error probability in decoding the interference versus the channel gain β. We observe that within 0.396dB of the very strong interference regime given by (IV.26) we are able to decode the interference with a bit error probability of less than 10^{-6}. Note that the main bottle neck in error performance in decoding the interference is the last i.e., the uncoded level whereas in decoding the desired signal (after the interference is decoded and subtracted), within $\sigma_{\max}$, arbitrarily small error rates can be achieved since no uncoded level needs to be decoded.

In Fig. IV.5, we plot the achievable rate as a function of P/σ^2 for the desired user for $r = 4$. It can be seen that the achievable rate with the lattice code has a gap of roughly 1.53 dB from the corresponding Shannon limit at high rates. This is

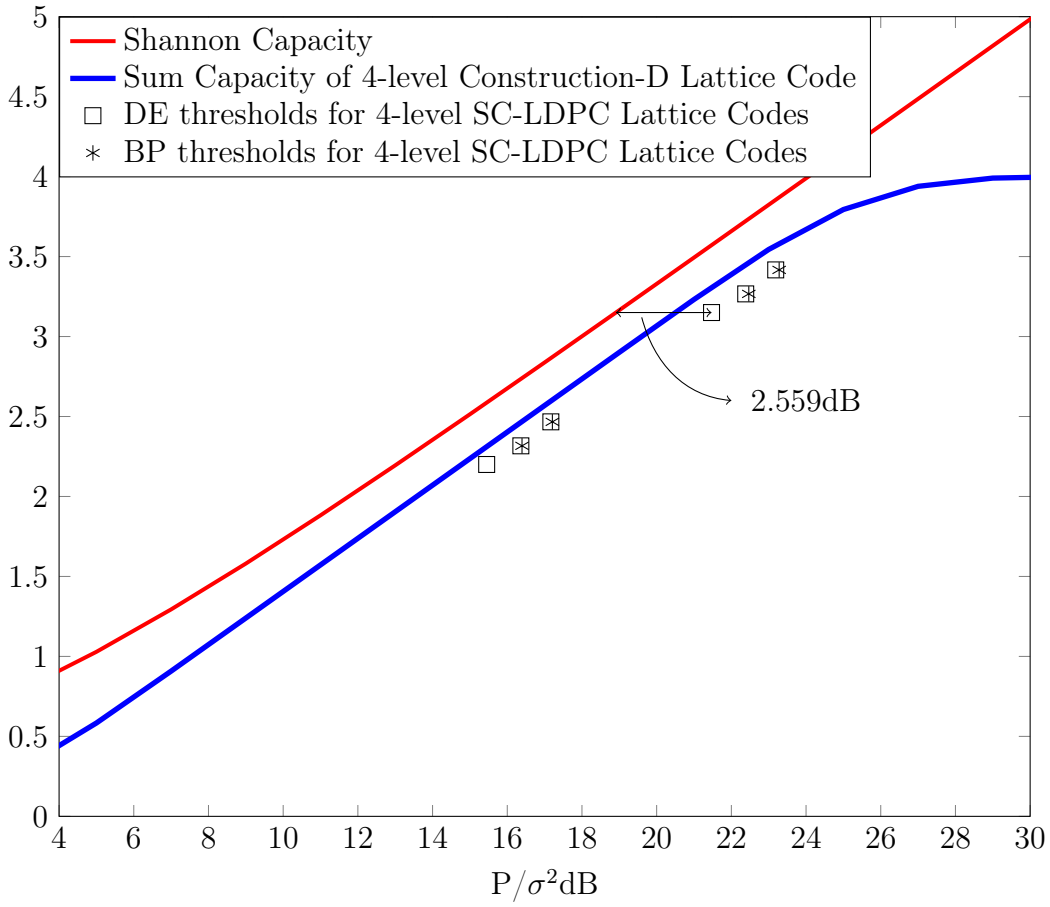

Figure IV.5: The gap between the Shannon capacity and the achievable sum-rate of a 4-level Construction-D lattice code(hypercube shaping) under multi-stage decoding. The DE thresholds, along with comparison with BP thresholds for $n = 2 \times 10^5$, for various SC-LDPC lattice codes with a maximum check node degree of 60 are also given.

the shaping loss due to hypercube shaping. The DE thresholds with the proposed SC-LDPC codes is also shown in the plot and it can be seen that the DE thresholds are very close to the achievable rates.

V. COMPRESSED SENSING*

V.A Introduction

The classical problem of compressed sensing involves estimating a signal $\mathbf{x}$, which is sparse in some basis, from a noisy measurement signal $\mathbf{y}$ of smaller dimension compared to $\mathbf{x}$. Formally, let

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w},$$

where $\mathbf{x}$ is an N-dimensional vector, $\mathbf{A}$ is a known $M \times N$ matrix commonly referred to as *measurement matrix* and $\mathbf{w}$ is additive noise. The unknown signal $\mathbf{x}$ is known to be sparse in some basis and we denote the sparsity of $\mathbf{x}$ by K. If there is no noise, then we refer to it as the noiseless setting. It is known that if $K \ll N$ we can recover the unknown signal in significantly fewer number of measurements compared to N. Particularly in this chapter we focus on recovering the support of $\mathbf{x}$ defined as $\text{supp}(\mathbf{x}) := \{i : x_i \neq 0, i \in [N]\}$ where $\mathbf{x} = [x_1, \ldots, x_i \ldots, x_N]^T$ and $[N] := \{1, 2, \ldots, N\}$. For a given scheme, given the reconstruction vector $\widehat{\mathbf{x}}$, we consider the probability of failure of support recovery which can be defined as

$$\mathbb{P}_F := \Pr(\text{supp}(\widehat{\mathbf{x}}) \neq \text{supp}(\mathbf{x})).$$

For the support recovery problem, under noisy settings, Wainwright [41] showed information theoretically that $O\left(K \log(\frac{N}{K})\right)$ number of measurements is necessary and sufficient for asymptotically reliable recovery.

In [16] (and in the expanded version in [4]), Li, Pawar and Ramchandran have considered the compressed sensing problem of recovering the support of a K-sparse, N-dimensional signal from M linear and noisy measurements. Based on sparse-graph codes with a *left-regular* degree profile and a peeling decoder, they have proposed an elegant design of the measurement matrix and a recovery algorithm. They have proposed two designs - the first design requires $M = \mathcal{O}(K \log N)$ measurements and a near-linear $\mathcal{O}(N \log N)$ decoding complexity, whereas the second design requires $M = \mathcal{O}(K \log N)$ measurements with a sub-linear $\mathcal{O}(K \log N)$ decoding complexity.

In this chapter[42], we show that the bounds on the measurement complexity reported in [16, 4] can be improved by considering *left-and- right-regular* sparse-graph based sensing matrices. We show that only $\mathcal{O}\left(K \log \frac{N}{K}\right)$ measurements are required when $K = \mathcal{O}(N^{\delta})$, for any $0 \leq \delta < 1$, to recover the support with the optimal sub-linear time decoding complexity. This matches the information-theoretic lower bound on the number of measurement required for asymptotically-reliable recovery [41]. Also, through simulations we demonstrate that the proposed scheme has superior performance compared to [4].

The literature on compressed sensing is vast and it is difficult to provide a comparison with several of the existing results in the literature due to different error performance metrics being used for different versions of the problem. Nevertheless, it should be pointed out that the use of left and right regular bipartite graphs as choice for sensing matrix has been proposed in [43] and the measurement complexity has been shown to be only $\mathcal{O}\left(K \log \frac{N}{K}\right)$. However, the decoding complexity is near-linear $\mathcal{O}(N \log \frac{N}{K})$. We achieve a similar measurement complexity but with optimal computational complexity of $\mathcal{O}(K \log \frac{N}{K})$. Also unlike in [43] the sensing matrix in this chapter is constructed based on a tensor-product construction and the decoding algorithm is based on identifying singletons and peeling them off.

For the information-theoretic lower bound in [41] to hold, the non-zero elements in $\mathbf{x}$ should have a sufficiently large minimum absolute value. In view of this condition, similar to [16, 4], we assume that all the non-zero elements of $\mathbf{x}$ belong to the set $\{Ae^{i\theta} : A \in \mathcal{A}, \theta \in \Theta\}$ where $\mathcal{A} := \{A_{\min} + \rho l\}_{l=0}^{L_1}, \Theta := \{2\pi l/L_2\}_{l=0}^{L_2}$ for finite but arbitrarily large integers L_1 and L_2.

V.B Prior work

In this section, we review the construction of the measurement matrix $\mathbf{A}$ proposed by Li, Pawar and Ramchandran in [16] and [4], and also summarize their key results. To keep the discussion simple, we omit certain details and refer readers to the original work [16] (and the expanded version [4]).

The measurement matrix is constructed using a combination of a sparse-graph code defined by the $R \times N$ coding matrix $\mathbf{H} = [\mathbf{h_1}, \mathbf{h_2}, \cdots, \mathbf{h_N}] \in \{0,1\}^{R \times N}$ and a $P \times N$ bin-detection matrix $\mathbf{S} = [\mathbf{s_1}, \mathbf{s_2}, \cdots, \mathbf{s_N}]$. The coding matrix $\mathbf{H}$ defines a bipartite graph $\mathcal{G}$ with N left (variable) nodes, representing the N-length signal $\mathbf{x}$, and R right (check) nodes representing the measurements $\mathbf{y} = [\mathbf{y_1}, \mathbf{y_2}, \cdots, \mathbf{y_R}]$. Let $q_i, i \in [R]$, be the number of non-zero variable nodes connected to i^{th} check node. Assume an "oracle" that solves the 1-sparse problem by examining each check node observations and classifies it as a zero-ton($q_i = 0$), singleton($q_i = 1$) or a multi-ton($q_i > 1$), and also identifies the position $\hat{k}$ and value $\widehat{x}_{\hat{k}}$ of the participating variable node if it is a single-ton. Once a singleton is identified the corresponding variable node's contribution is peeled off from other participating check nodes and this process creates new single-tons. The decoding process continues until there are no more singletons. The decoding is successful if all the K non-zero elements of $\mathbf{x}$ are recovered at the end of decoding.

The $RP \times N$ measurement matrix $\mathbf{A}$ with $M = RP$ measurements is constructed

by taking the row tensor product $\boxtimes$ of $\mathbf{H}$ and $\mathbf{S}$ given by

$$\mathbf{A} = \mathbf{H} \boxtimes \mathbf{S} := [\mathbf{h_1} \otimes \mathbf{s_1}, \mathbf{h_2} \otimes \mathbf{s_2}, \cdots, \mathbf{h_N} \otimes \mathbf{s_N}]$$

where $\otimes$ is the standard Kronecker product.

For the noisy setting, they have proposed three designs for $\mathbf{S}$ which essentially performs the role of the oracle in identifying a single-ton at each check node. These designs are *RandomNoisy* with near-linear decoding complexity, *BinaryNoisy* and *FourierNoisy* each with sub-linear decoding complexity. The bin-detection matrix $\mathbf{S}$ for the three settings are as follows:

- *RandomNoisy:* Ensemble of $P \times N$ matrices $S = [S_{i,j}]_{P \times N}$ where $S_{i,j}$ s are i.i.d. sub-gaussian entries with zero mean and unit variance.
- *FourierNoisy*: $\mathbf{S} = [\mathbf{S_0 S_1} \cdots \mathbf{S_{P-1}}]^{\mathbf{T}}$, where $\mathbf{S_p}$ consists of $Q = O(\log^{1/3} N)$ consecutive 2^p-dyadically spaced rows from the $N \times N$ DFT matrix.
- *BinaryNoisy:* $\mathbf{S} = f(\mathbf{C})$ where $\mathbf{C}_{P \times N}$ is a binary codebook(or subset of a codebook) of a linear code with block length P, $f : \{0,1\}^q \to \mathcal{M}$ is a modulation scheme that maps $C_{P \times N}$ to $S_{\frac{P}{q} \times N}$. For e.g., for QAM $q = 2$ and $\mathcal{M} = \{\pm 1 \pm i\}$.

The following theorems from [4] summarize their key results.

Theorem 52 ([4] Sub-linear Time Noisy Recovery)**.** In the presence of i.i.d. Gaussian noise with zero mean and variance σ^2, given any K-sparse signal $\mathbf{x}$ with $x_k \in \mathcal{X}$ for $k \in \text{supp}(\mathbf{x})$, our noiseless recovery schemes achieve a vanishing failure probability $\mathbb{P}_F \to 0$ asymptotically in K and N with

	M	T
Fourier noisy	$O(K \log^{1.3} N)$	$O(K \log^{1.3} N)$
Binary noisy	$O(K \log N)$	$O(K \log N)$

where M and T are measurement cost and computational complexity respectively.

Theorem 53 ([4] Near-linear Time Noisy Recovery)**.** In the presence of i.i.d. Gaussian noise with zero mean and variance σ^2, given any K-sparse signal $\mathbf{x}$ with $x_k \in \mathcal{X}$ for $k \in \text{supp}(\mathbf{x})$, the RandomNoisy scheme achieves a vanishing failure probability $\mathbb{P}_F \to 0$ asymptotically in K and N with a measurement complexity of $M = O(K \log N)$ and computational complexity of $T = O(N \log N)$.

V.C Proposed scheme

The main difference between [4] and our approach is that we replace the left l-regular ensemble of graphs corresponding to the coding matrix $\mathbf{H}$ described in Sec V.B by left *and* right (l, r)-regular ensemble of graphs.

Definition 54 (Left and right regular graph ensemble)**.** Let $\mathcal{G}^N_{\text{reg,reg}}\left(R, l, \frac{lN}{R}\right)$ denote the ensemble of left and right regular bipartite graphs with N variable nodes and R check nodes, where each variable node $k \in [N]$ is connected to l check nodes and each check node $j \in [R]$ is connected to $\frac{lN}{R}$ left nodes.

In the design considerations of bin detection matrix, we now have only $r = \frac{lN}{\eta K} = O\left(\frac{N}{K}\right)$ variable nodes connected to each check node and thus we require only a bin detection matrix $\mathbf{S}$ with $O(\frac{N}{K})$ columns. For the bin detection matrix designs in Sec. V.B we know from [4] that to differentiate between a zero-ton, singleton and a multi-ton successfully with probability approaching 1 asymptotically in $\frac{N}{K}$ we only require $\log(\frac{N}{K})$ rows in $\mathbf{S}$. We choose the bin detection matrix to be similar to the *RandomNoisy, FourierNoisy, BinaryNoisy* designs but with dimensions $P' \times r$ where $P' = O\left(\log(\frac{N}{K})\right)$.

We know from the modern coding theory that to peel off K unknown variable nodes successfully from the bipartite graph we need ηK number of check nodes for

some $\eta > 1$. So we choose the number of check nodes $R = \eta K$. A matrix $\mathbf{H}$ is chosen at random from this ensemble $\mathcal{G}^{N}_{\text{reg,reg}}\left(\eta K, l, \frac{lN}{\eta K}\right)$ and used as the coding matrix.

The measurement matrix $\mathbf{A}$ for the proposed construction with $\mathbf{H_{R\times N}} = [\mathbf{h_1}, \mathbf{h_2}, \cdots, \mathbf{h_N}]^{\mathbf{T}}$ and $\mathbf{S_{P'\times r}} = [\mathbf{s_1}, \mathbf{s_2}, \cdots, \mathbf{s_r}]$ is given by $\mathbf{A}_{RP'\times N} = \mathbf{H} \boxplus \mathbf{S}$, where $\boxplus$ is the new tensoring operation, which is slightly different from the row-tensor operation used in Sec V.B and is defined as

$$\mathbf{A}_{RP'\times N} = \mathbf{H} \boxplus \mathbf{S} = \begin{bmatrix} \mathbf{h_1} \boxtimes \mathbf{S_1} \\ \mathbf{h_2} \boxtimes \mathbf{S_2} \\ \vdots \\ \mathbf{h_R} \boxtimes \mathbf{S_R} \end{bmatrix}$$

where,

$\mathbf{S_i} = [\mathbf{0}, \cdots, \mathbf{s_1}, \mathbf{0}, \cdots, \mathbf{s_2}, \cdots, \mathbf{0}, \mathbf{s_r}, \cdots, \mathbf{0}]$, $(i \in [R])$, where $\mathbf{0}$ is an all-zero column vector of length P' placed in positions j where $h_{ij} = 0$ and the column vectors $\mathbf{s_k}$, $k \in [r]$ are placed sequentially in the positions j where $h_{ij} = 1$. We illustrate the new tensoring operation $\boxplus$ via Example 55.

Example 55. Let

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

denote an adjacency matrix from the ensemble $\mathcal{G}^{6}_{\text{reg,reg}}(4, 2, 3)$. We choose $P' =$

$\lceil \log_2 r \rceil = 2$ and let $S_{P' \times r}$ be defined as

$$S = \begin{bmatrix} +1 & -1 & -1 \\ -1 & +1 & -1 \end{bmatrix}$$

Then, the measurement matrix $\mathbf{A}$ with $M = P'R = 8$ measurements is given by

$$A = H \boxplus S \; = \begin{bmatrix} +1 & 0 & 0 & -1 & 0 & -1 \\ -1 & 0 & 0 & +1 & 0 & -1 \\ 0 & +1 & -1 & 0 & -1 & 0 \\ 0 & -1 & +1 & 0 & -1 & 0 \\ +1 & -1 & 0 & -1 & 0 & 0 \\ -1 & +1 & 0 & -1 & 0 & 0 \\ 0 & 0 & +1 & 0 & -1 & -1 \\ 0 & 0 & -1 & 0 & +1 & -1 \end{bmatrix}$$

V.D Improved bounds

With our proposed construction of the measurement matrix, Theorem 52 and Theorem 53 can be sharpened to the following new theorems.

Theorem 56 (Sub-linear Time Noisy Recovery)**.** In the presence of i.i.d. Gaussian noise with zero mean and variance σ^2, given any K-sparse signal $\mathbf{x}$ with $x_k \in \mathcal{X}$ for $k \in \text{supp}(\mathbf{x})$, our noisy recovery schemes achieve a vanishing failure probability $\mathbb{P}_F \to 0$ asymptotically in K and N with

	M	T
Fourier noisy	$O\left(K \log^{1.3} \frac{N}{K}\right)$	$O\left(K \log^{1.3} \frac{N}{K}\right)$
Binary noisy	$O\left(K \log \frac{N}{K}\right)$	$O\left(K \log \frac{N}{K}\right)$

Theorem 57 (Near-linear Time Noisy Recovery)**.** In the presence of i.i.d. Gaussian noise with zero mean and variance σ^2, given any K-sparse signal $\mathbf{x}$ with $x_k \in \mathcal{X}$ for $k \in \text{supp}(\mathbf{x})$, the RandomNoisy scheme achieves a vanishing failure probability $\mathbb{P}_F \to 0$ asymptotically in K and N with a measurement complexity of $M = O\left(K \log \frac{N}{K}\right)$ and computational complexity of $T = O\left(N \log \frac{N}{K}\right)$.

Proof. The bin detection matrix and the decoding methods employed to identify a singleton are identical to that of [4] except that $P = O(\log N)$ is replaced by $P' = O\left(\log(\frac{N}{K})\right)$. Hence the probability of error for the bin detection algorithm can be analyzed exactly as in [4] with P' replaced by P and thus can be shown to be exponentially decaying in P'. For this particular choice of P' the probability of error for the bin detection part vanishes asymptotically in $\frac{N}{K}$. Therefore for K sub-linear in N all it remains to be shown is that the $\mathcal{G}^N_{\text{reg,reg}}\left(R, l, \frac{lN}{R}\right)$ ensemble with peeling process fails with a vanishing error probability $\mathbb{P}_F$ asymptotically in K and N. For choice of $l \geq 3$, Theorem 64 gives us this required result and that completes the proof. □

V.E Proofs

In this section we consider a $\mathcal{G}^N_{\text{reg,reg}}(R, l, \frac{lN}{R})$ ensemble and show that this ensemble with the oracle based peeling decoder fails to recover all the variable nodes with a probability of at most $O\left(\frac{1}{K}\right)$. Although it appears this can be achieved directly by using a capacity achieving spatially-coupled LDPC ensemble and use the existing results, there are two main obstacles to this:

- In traditional LDPC codes and peeling decoder over binary erasure channel, the input to the decoder is the channel output corresponding to N variable (bit) nodes and the check nodes on the right are mere parity checks whose sum modulo 2 is zero. Whereas in our problem the values corresponding to the

N variable nodes on the left need to be evaluated by the decoder given the values corresponding to the R check nodes (the real sum of the variable nodes connected) are non-zero and form input to the decoder.

- In traditional LDPC case a constant fraction ϵ of these N variable nodes are erased by the channel and usually the emphasis is on analyzing the performance of peeling decoder asymptotically in N or R when rate$=1-\frac{R}{N}$ is fixed. But in our case the fraction of the nodes erased $= 1-\frac{K}{N}$, where K, sub-linear in N, is usually of the form $K = N^{\delta}$, tend to one and the rate of the code$= 1-\frac{R}{N} = 1-\frac{\eta N^{\delta}}{N}$ tend to one asymptotically in N.

Consider a left and right regular LDPC code $\mathcal{G}_{\text{LDPC}}(N, l, r)$ where N is the number of variable nodes on the left and l, r are the regular left and right degrees respectively. Let $\mathrm{P}_{\text{BEC}}^{(i)}(\mathbf{y})$ be the degree distribution of the number of check nodes after iteration i of peeling decoder given $\mathbf{y}$ is the channel output. And similarly $\mathcal{G}_{\text{reg,reg}}^{N}(R, l, \frac{lN}{R})$ be the graph corresponding to the parity check matrix in the support recovery problem and $\mathrm{P}_{\text{SR}}^{(i)}(\mathbf{z})$ be the degree distribution of the check nodes on the right after iteration i of the oracle-based peeling decoder, given $\mathbf{z}$ is the support recovery equivalent of syndrome corresponding to $\mathbf{x}$ i.e., $\mathbf{z} = \mathbf{H}\mathbf{x}$ where the operations are over the real field.

Note that in the peeling decoder, we peel off one degree-1 check node and the variable node connected to it from the graph in each iteration. In the LDPC-BEC problem we remove all the variable nodes that are not erased by the channel and the resulting graph is input to the decoder. Similarly in the case of support recovery problem we consider the oracle based peeling decoder in [4] and we analyze the *pruned*-graph where we remove all the zero variable nodes from the original graph and input to the decoder.

Lemma 58 (Equivalence to LDPC-BEC)**.** Whenever $\mathbf{y}$ and $\mathbf{z}$ satisfy

$$\mathbf{z} = \mathbf{H}\mathbf{x} \text{ such that } S := |\text{supp}(\mathbf{x})| = |\{i : y_i = \mathcal{E}\}|$$

where $\mathcal{E}$ denotes erasure, then $\mathrm{P}^{(i)}_{\mathrm{BEC}}(\mathbf{y}) = \mathrm{P}^{(i)}_{\mathrm{SR}}(\mathbf{z}) \quad \forall i$.

Proof. Define $S^c = [1 : N] \backslash S$. In the case of LDPC codes on BEC we peel off all non-erased variable nodes corresponding to S^c and input the resulting graph to the peeling decoder. Similarly in the case of bipartite graph in support recovery problem we peel off all the zero nodes corresponding to S^c and we input the resulting graph to oracle based peeling decoder. From this point onward the peeling decoders are identical and thus we have our result. □

Thus by considering a BEC of erasure probability $\epsilon = \frac{K}{N}$ we can equivalently consider peeling decoder of LDPC codes on BEC channel and use various existing results.

Lemma 59. The evolution of the left and right degree distribution as the peeling decoder progresses can be given by

$$\begin{aligned}
\tilde{L}_l(y) &= y^k l, \\
\tilde{R}_1(y) &= r\epsilon y^{l-1}[y - 1 + (1 - \epsilon y^{l-1})^{r-1}] \\
\tilde{R}_i(y) &= \binom{r}{i}(\epsilon y^{l-1})^i (1 - \epsilon y^{l-1})^{r-1}, \quad i \geq 2
\end{aligned}$$

where $\epsilon = \frac{K}{N}$ and $r = \frac{lN}{\eta K}$. Note that the curve corresponding to $\tilde{L}_i(y)(\tilde{R}_i(y))$ for $y \in [0, 1]$ gives the expected number of degree i variable nodes (check nodes) normalized with respect to K (ηK).

Proof. As we showed in Lemma. 58 the peeling decoder for an LDPC on BEC channel and oracle based peeling decoder for CS are identical upto the residual degree distributions at each iteration. Hence we can use the result for LDPC codes [22, Theorem 3.107] with equivalent channel erasure probability $\epsilon = \frac{K}{N}$. □

Definition 60 (BP Threshold)**.** We define the BP threshold, η^{BP} to be the minimum value of η for which there is no non-zero solution for the equation:

$$\begin{aligned} y &= \lim_{\frac{N}{K}\to\infty} 1 - \left(1 - \frac{Ky^{l-1}}{N}\right)^{\frac{lN}{\eta K}} \\ &= 1 - e^{\frac{-ly^{l-1}}{\eta}} \end{aligned}$$

in the range $y \in [0, 1]$.

Lemma 61. [22, Theorem 3.107] If $\eta > \eta^{BP}$ then with probability at least $1 - O\left(K^{1/6} e^{-\frac{\sqrt{Kl}}{(lr)^3}}\right)$ the peeling decoder of a specific instance progresses until the number of residual variable nodes in the graph has reached size γK where γ is an arbitrary positive constant.

Definition 62 (Expander Graphs)**.** A bipartite graph with K left nodes and regular left degree l is called a $(\gamma, 1/2)-$ expander if for all subsets S of left nodes with $|S| \leq \gamma K$, the right neighborhood of S denoted by $\mathcal{N}(S)$ satisfies $|\mathcal{N}(S)| > l|S|/2$.

Lemma 63. Consider a left and right regular ensemble $\mathcal{G}^N_{\text{reg,reg}}(\eta K, l, \frac{Nl}{\eta K})$, then the pruned graph resulting from any given K-sparse signal $\mathbf{x}$ is a $(\gamma, 1/2)$-expander with probability at least $1 - O\left(\frac{1}{K^{l-2}}\right)$ for a sufficiently small constant $\gamma > 0$.

Proof. The proof is similar to the proof used in [4] with minor modifications. Let E_v denote the event that a subset S_v of variable nodes on the left with size v has at

most $l|S_v|/2$ neighbors whose probability can be computed as

$$\Pr(E_v) \leq \binom{K}{v}\binom{\eta K}{lv/2}\left(\frac{vl}{2\eta K}\right)^{lv} \tag{V.1}$$

$$\leq c^{vl/2}\left(\frac{v}{K}\right)^{v(l/2-1)} \tag{V.2}$$

where $c = \frac{le^2}{2\eta}$ is a constant. In (V.1) we upper bound the probability of E_v via union bound over all possible size v subsets on the left and size $lv/2$ subsets on the right. In (V.2) we use the inequality $\binom{a}{b} \leq (ae/b)^b$ and we assume $l \geq 2$ to simplify the constant factor. Then we union bound over all subsets of size upto the remaining nodes $\gamma^* K$ where we choose $\gamma^* = \left(4c^l\right)^{\frac{-1}{l-2}}$

$$\sum_{v=2}^{\gamma^* K} \Pr(E_v) \leq \sum_{v=2}^{\gamma^* K}\left(c^l\left(\frac{v}{K}\right)^{l-2}\right)^{v/2}$$

$$= O\left(\frac{1}{K^{l-2}}\right)$$

Thus we showed that asymptotically in K, the left and right regular graphs are good expander graphs with probability atleast $1 - O(1/K^{l-2})$. □

Theorem 64. Consider the ensemble $\mathcal{G}^N_{\text{reg-reg}}(\eta K, l, \frac{Nl}{\eta K})$, the oracle based peeling decoder peels off all the variable nodes in the pruned graph in ηK iterations with probability at least $1 - O\left(1/K^{l-2}\right)$.

Proof. Lemma 61 shows us that the peeling decoder fails to peel off till the residual graph has γN variable nodes remaining with an exponentially low probability. Then in Lemma 63 we show that the left regular graphs are good expanders with a probability of atleast $1 - O(1/K^{l-2})$ and hence the remaining γN nodes can be peeled off with high probability. Thus the overall probability of failure will be dominated by

small stopping sets which can be upper bounded by $O(1/K^{l-2})$. $\square$

V.F Numerical results

In this section we provide the empirical performance of our scheme in the noisy setting. We fix the parameters $K = 50$ and $N = 10^5$. For a given SNR we generate a K-sparse signal at random and perform the support recovery for this signal over 200 sensing matrices sampled from the proposed construction. Specifically, $\text{supp}(\mathbf{x})$ is chosen uniformly at random from $[N]$ and the non-zero values in $\mathbf{x}$ are chosen uniformly at random from the set $\{+1, -1\}$. We sample the coding matrix $\mathbf{H}$ from the ensemble $\mathcal{G}^N_{\text{reg,reg}}(R = 2K, l = 4, r = \frac{2N}{K})$ for each simulation. For the bin detection matrix we consider the *BinaryNoisy* scheme and we use two classes of codes: convolutional codes and (12,24) Golay code with QAM modulation. In the case of convolutional codes we consider $(12, n)$ truncated convolutional code corresponding to rates $\frac{1}{2}, \frac{1}{4}$ and $\frac{1}{8}$ with a constraint length of 8 which results in $n = 24, 48$ and 96 respectively. This gives bin detection matrix dimensions of $12 \times r, 24 \times r$ and $48 \times r$ where $r = 4000$ is the right degree of the graph corresponding to $\mathbf{H}$. For the singleton identification Viterbi soft decision decoding is considered for convolutional codes whereas a hard decision syndrome decoding is considered for Golay code resulting in a decoding complexity of $O\left(K \log\left(\frac{N}{K}\right)\right)$. We observe from Fig. V.1 that the Golay code based construction with $M = 1300$ has similar performance and the convolutional code based construction with $M = 2400$ has better performance when compared to that of $M = 9600$ *BinaryNoisy* scheme with sub-linear time complexity decoder of LPR[16].

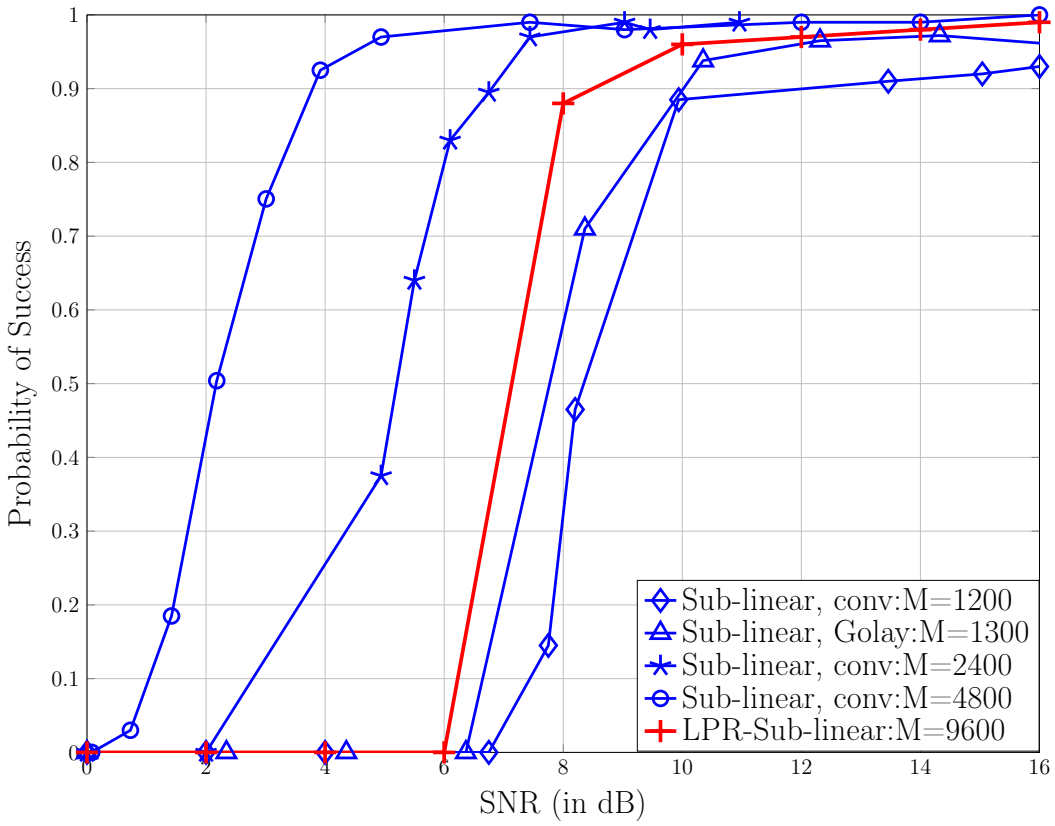

Figure V.1: Probability of Success for our construction (blue curves) with *BinaryNoisy* scheme using convolutional codes(conv) and Golay code with sub-linear time decoding complexity of $O(K \log \frac{N}{K})$.And we compare the performance with that of *BinaryNoisy* scheme by Li, Pawar and Ramachandran (LPR) (red curve) [4] with sub-linear decoding complexity of $O\left(K \log N\right)$.

V.G Conclusion

In this chapter we considered the support recovery problem in compressed sensing and proposed a sensing matrix construction based on left-and-right regular sparse-graph ensemble. It was shown that the proposed construction, using an order optimal measurement complexity of $\mathcal{O}(K \log \frac{N}{K})$, recovers the support of the sparse signal with asymptotically vanishing error probability in optimal sub-linear time complexity of $\mathcal{O}(K \log \frac{N}{K})$.

VI. GROUP TESTING

VI.A Introduction

The problem of Group Testing (GT) refers to testing a large population of N items for K defective items (or sick people) where grouping multiple items together for a single test is possible. The output of the test is *negative* if all the grouped items are non-defective or else the output is *positive.* In the scenario when $K \ll N$, the objective of GT is to design the testing scheme such that the total number of tests m to be performed is minimized.

This problem was first introduced to the field of statistics by Dorfman [17] during World War II for testing the soldiers for syphilis without having to test each soldier individually. Since then group testing has found application in wide variety of problems like clone library screening, non-linear optimization, multi-access communication etc.., [44] and fields like biology[45], machine learning[46], data structures[47] and signal processing[48]. A comprehensive survey on group testing algorithms, both combinatorial and probabilistic, can be found in [44, 49, 50].

In the literature on Group Testing, three kinds of reconstruction guarantees have been considered: combinatorial, probabilistic and approximate. In the combinatorial designs for the GT problem, the probability of recovery for any given defective set should be equal to 1 whereas in the probabilistic version one is interested in recovering *all* the defective items with high probability (w.h.p) i.e., with probability approaching 1 asymptotically in N and K. Another variant of the probabilistic version is that the probability of recovery is required to be greater than or equal to $(1-\varepsilon)$ for a given $\varepsilon > 0$. For the approximate recovery version one is interested in only recovering a $(1-\epsilon)$ fraction of the defective items (not the whole set of defective items) w.h.p.

For the combinatorial GT the best known lower bound on the number of tests required is $\Omega(K^2\frac{\log N}{\log K})$ [51, 52] whereas the best known achievability bound is $\mathcal{O}(K^2 \log N)$ [53, 54]. Most of these results were based on algorithms relying on exhaustive searches thus have a high computational complexity of atleast $\mathcal{O}(K^2 N \log N)$. Only recently a scheme with efficient decoding was proposed by Indyk et al., [55] where all the defective items are guaranteed to recover using $m = \mathcal{O}(K^2 \log N)$ tests in $\text{poly}(K) \cdot \mathcal{O}(m \log^2 m) + \mathcal{O}(m^2)$ time.

If we consider the probabilistic version of the problem, it was shown in [49, 50] that the number of tests necessary is $\Omega(K \log \frac{N}{K})$ which is the best known lower bound in the literature. And regarding the best known achievability bound Mazumdar [56] proposed a construction that has an asymptotically decaying error probability with $\mathcal{O}(K\frac{\log^2 N}{\log K})$ tests. For the approximate version it was shown [50] that the required number of tests scale as $\mathcal{O}(K \log N)$ and to the best of our knowledge this is the tightest bound known.

In [5] authors Lee, Pedarsani and Ramchandran proposed a testing scheme based on *left-regular sparse-graph* codes and a simple iterative decoder based on the*peeling* decoder, which are popular tools in channel coding [22], for the non-adaptive group testing problem. They refer to the scheme as SAFFRON(**S**parse-gr**A**ph codes **F**ramewrok **F**or g**RO**up testi**N**g), a reference which we will follow through this document. The authors proved that using SAFFRON scheme $m = c_\epsilon K \log N$ number of tests are enough to identify atleast $(1 - \epsilon)$ fraction of defective items (the approximate version of GT) w.h.p. The precise value of constant c_ϵ as a function of the required error floor ϵ is also given. More importantly the computational complexity of the proposed peeling based decoder is only $\mathcal{O}(K \log N)$. They also showed that with $m = c \cdot K \log K \log N$ tests i.e. with an additional $\log K$ factor, the *whole* defective set (the probabilistic version of GT) can be recovered with an asymptotically high

probability of $1 - \mathcal{O}(K^{-\alpha})$.

Our contributions

In this work[57], we propose a non-adaptive GT scheme that is similar to the SAFFRON but we employ *left-and-right-regular sparse-graph* codes instead of the left-regular sparse-graph codes and show that we only require $c_\epsilon K \log \frac{N\ell}{c_\epsilon K}$ number of tests for an error floor of ϵ in the approximate version of the GT problem. Although the testing complexity of our scheme has the same asymptotic order $\mathcal{O}(K \log N)$ as that of [5], which as far as we are aware is the best known order result for the required number of tests in the approximate GT, it provides a better explicit upper bound of $\Theta(K \log \frac{N}{K})$ with optimal computational complexity $\mathcal{O}(K \log \frac{N}{K})$ and also a significant improvement in the required number of tests for finite values of K, N. Following the approach in [5] we extend our proposed scheme with the singleton-only variant of the decoder to tackle the probabilistic version of the GT problem. In Sec. VI.E we show that for $m = c \cdot K \log K \log \frac{N}{K}$ tests i.e. with an additional $\log K$ factor the *whole* defective set can be recovered w.h.p. Note that the testing complexity of our scheme is only $\log K$ factor away from the best known lower bound of $\Omega(K \log \frac{N}{K})$ [49] for the probabilistic GT problem. We also extend our scheme to the noisy GT problem, where the test results are corrupted by noise, using an error-correcting code similar to the approach taken in [5]. We demonstrate the improvement in the required number of tests due to *left*-and-*right*-regular graphs for finite values of K, N via simulations.

VI.B Problem statement

Formally the group testing problem can be stated as following. Given a total number of N items out of which K are defective, the objective is to perform m different tests and identify the location of the K defective items from the test outputs. For now we consider only the noiseless group testing problem i.e., the result of each

test is exactly equal to the boolean OR of all the items participating in the test.

Let the support vector $\mathbf{x} \in \{0,1\}^N$ denote the list of items in which the indices with non-zero values correspond to the defective items. A non-adaptive testing scheme consisting of m tests can be represented by a matrix $\mathbf{A} \in \{0,1\}^{m \times N}$ where each row $\mathbf{a}_i$ corresponds to a test. The non-zero indices in row $\mathbf{a}_i$ correspond to the items that participate in i^{th} test. The output corresponding to vector $\mathbf{x}$ and the testing scheme $\mathbf{A}$ and can be expressed in matrix form as:

$$\mathbf{y} = \mathbf{A} \odot \mathbf{x}$$

where $\odot$ is the usual matrix multiplication in which the arithmetic multiplications are replaced by the boolean AND operation and the arithmetic additions are replaced by the boolean OR operation.

VI.C Review: SAFFRON

As mentioned earlier the SAFFRON scheme [5] is based on left-regular sparse graph codes and is applied for non-adaptive group testing problem. In this section we will briefly review their testing scheme, iterative decoding scheme (reconstruction of $\mathbf{x}$ given $\mathbf{y}$) and their main results. The SAFFRON testing scheme consists of two stages: the first stage is based on a left-regular sparse graph code which pools the N items into M non-disjoint bins where each item belongs to exactly ℓ bins. The second stage comprises of producing h testing outputs at each bin where the h different combinations of the pooled items (from the first stage) at the respective bin are defined according to a universal signature matrix. For the first stage the authors consider a bipartite graph with N variable nodes (corresponding to the N items) and M bin nodes. Each variable node is connected to ℓ bin nodes chosen uniformly at random from the M available bin nodes. All the variable nodes (historically depicted

on the left side of the graph in coding theory) have a degree ℓ, hence the left-regular, whereas the degree of a bin node on the right is a random variable in the range $[0 : N]$.

Definition 65 (Left-regular sparse graph ensemble)**.** Let $\mathcal{G}_\ell(N, M)$ be the ensemble of left-regular bipartite graphs where for each variable node the ℓ right node connections are chosen uniformly at random from the M right nodes.

Let $\mathbf{T}_G \in \{0,1\}^{M\times N}$ be the adjacency matrix corresponding to a graph $G \in \mathcal{G}_\ell(N, M)$ i.e., each column in $\mathbf{T}_G$ corresponds to a variable node and has exactly ℓ ones. Let the rows in matrix $\mathbf{T}_G$ be given by $\mathbf{T}_G = [\mathbf{t}_1^T, \mathbf{t}_2^T, \ldots, \mathbf{t}_M^T]^T$. For the second stage let the universal signature matrix defining the h tests at each bin be $\mathbf{U} \in \{0,1\}^{h\times N}$. Then the overall testing matrix $\mathbf{A} := [\mathbf{A}_1^T, \ldots, \mathbf{A}_M^T]^T$ where $\mathbf{A}_i = \mathbf{U}\,\mathrm{diag}(\mathbf{t}_i)$ of size $h \times N$ defines the h tests at i^{th} bin. Thus the total number of tests is $m = M \times h$.

The signature matrix $\mathbf{U}$ in a more general setting with parameters r and p can be given by

$$\mathbf{U}_{r,p} = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \cdots & \mathbf{b}_r \\ \overline{\mathbf{b}}_1 & \overline{\mathbf{b}}_2 & \cdots & \overline{\mathbf{b}}_r \\ \mathbf{b}_{\pi_1^1} & \mathbf{b}_{\pi_2^1} & \cdots & \mathbf{b}_{\pi_r^1} \\ \overline{\mathbf{b}}_{\pi_1^1} & \overline{\mathbf{b}}_{\pi_2^1} & \cdots & \overline{\mathbf{b}}_{\pi_r^1} \\ \cdots & & \vdots & \\ \mathbf{b}_{\pi_1^{p-1}} & \mathbf{b}_{\pi_2^{p-1}} & \cdots & \mathbf{b}_{\pi_r^{p-1}} \\ \overline{\mathbf{b}}_{\pi_1^{p-1}} & \overline{\mathbf{b}}_{\pi_2^{p-1}} & \cdots & \overline{\mathbf{b}}_{\pi_r^{p-1}} \end{bmatrix} \tag{VI.1}$$

where $\mathbf{b}_i \in \{0,1\}^{\lceil \log_2 r\rceil}$ is the binary expansion vector for i and $\overline{\mathbf{b}}_i$ is the complement of $\mathbf{b}_i$. $\pi^k = [\pi_1^k, \pi_2^k, \ldots, \pi_r^k]$ denotes a permutation chosen at random from symmetric

group S_r. Henceforth $\mathbf{U}_{r,p}$ will refer to either the ensemble of matrices generated over the choices of the permutations π^k for $k \in [1 : p-1]$ or a matrix picked uniformly at the random from the said ensemble. The reference should be sufficiently clear from the context. In the SAFFRON scheme the authors employed a signature matrix from $\mathbf{U}_{r,p}$ with $r = N$ and $p = 3$ thus resulting in a $\mathbf{U}$ of size $h \times N$ with $h = 6 \log_2 N$.

Decoding

Before describing the decoding process let us review some terminology. A bin is referred to as a *singleton* if there is exactly one non-zero variable node connected to the bin and similarly referred to as a *double-ton* in case of two non-zero variable nodes. In the case where we know the identity of one of them leaving the decoder to decode the identity of the other one, the bin is referred to as a *resolvable double-ton*. And if the bin has more than two non-zero variable nodes attached we refer to it as a *multi-ton*. First part of the decoder which is referred to as bin decoder will be able to detect and decode exactly the identity of the non-zero variable nodes connected to the bin if and only if the bin is a singleton or a resolvable double-ton. If the bin is a multi-ton the bin decoder will detect it neither as a singleton nor a resolvable double-ton with high probability. The second part of the decoder which is commonly referred to as peeling decoder [16], when given the identities of some of the non-zero variable nodes by the bin decoder, identifies the bins connected to the recovered variable nodes and looks for newly uncovered resolvable double-ton in these bins. This process of recovering new non-zero variable nodes from already discovered non-zero variable nodes proceeds in an iterative manner (referred to as peeling off from the graph historically). For details of the decoder we refer the reader to [5].

The overall group testing decoder comprises of these two decoders working in

conjunction as follows. In the first and foremost step, given the m tests output, the bin decoder is applied on the M bins and the set of variable nodes that are connected to singletons are decoded and output. We denote the decoded set of non-zero variable nodes as $\mathcal{D}$. Now in an iterative manner, at each iteration, a variable node from $\mathcal{D}$ is considered and the bin decoder is applied on the bins connected to this variable node. The main idea is that if one of these bins is detected as a resolvable double-ton thus resulting in decoding a new non-zero variable node. The considered variable node in the previous iteration is moved from $\mathcal{D}$ to a set of peeled off variable nodes $\mathcal{P}$ and the newly decoded non-zero variable node in the previous iteration, if any, will be placed in set $\mathcal{D}$ and continue to the next iteration. The decoder is terminated when $\mathcal{D}$ is empty and is declared successful if the set $\mathcal{P}$ equals the set of defective items.

Remark 66. Note that we are not literally peeling off the decoded nodes from the graph because of the *non-linear* OR operation on the non-zero variable nodes at each bin thus preventing us in subtracting the effect of the non-zero node from the measurements of the bin node unlike in the problems of compressed sensing or LDPC codes on binary erasure channel.

Now we state the series of lemmas and theorems from [5] that enabled the authors to show that their SAFFRON scheme with the described peeling decoder solves the group testing problem with $c \cdot K \log N$ tests and $\mathcal{O}(K \log N)$ computational complexity.

Lemma 67 (Bin decoder analysis)**.** For a signature matrix $\mathbf{U}_{r,p}$ as described in (VI.1), the bin decoder successfully detects and resolves if the bin is either a singleton or a resolvable double-ton. In the case of the bin being a multi-ton, the bin decoder declares a wrong hypothesis of either a singleton or a resolvable double-ton with a

probability no greater than $\frac{1}{r^{p-1}}$.

Proof. This result was proved in [5] for the choice of parameters $r = N$ and $p = 3$. The extension of the result to general r, p parameters is straight forward. □

For convenience the performance of the peeling decoder is analyzed independently of the bin decoder i.e., a peeling decoder is considered which assumes that the bin decoder is working accurately which will be referred to as *oracle based peeling decoder*. Another simplification is that a pruned graph is considered where all the zero variable nodes and their respective edges are removed from the graph. Also the oracle based peeling decoder is assumed to decode a variable node if it is connected to a bin node with degree one or degree two with one of them already decoded, in an iterative fashion. Any right node with more than degree two is untouched by this oracle based peeling decoder. It is easy to verify that the original decoder with accurate bin decoding is equivalent to this simplified oracle based peeling decoder on a pruned graph.

Definition 68 (Pruned graph ensemble)**.** Let the pruned graph ensemble $\tilde{\mathcal{G}}_l(N, K, M)$ be the set of all bipartite graphs obtained from removing a random $N - K$ subset of variable nodes from a graph from the ensemble $\mathcal{G}_\ell(N, M)$. Note that graphs from the pruned ensemble have K variable nodes.

Before we analyze the pruned graph ensemble let us define the right-node degree distribution (d.d) of an ensemble as $R(x) = \sum_i R_i x^i$ where R_i is the probability that a right-node in any graph from the ensemble has degree i. Similarly the edge d.d $\rho(x) = \sum_i \rho_i x^{i-1}$ is defined where ρ_i is the probability that a random edge in the graph is connected to a right-node of degree i. Note that the left-degree distribution is regular (i.e. $L(x) = x^\ell$) even for the pruned graph ensemble and hence is not specifically discussed.

Lemma 69 (Edge d.d of Pruned graph)**.** For the pruned ensemble $\tilde{\mathcal{G}}_\ell(N, K, M)$, it was shown that in the limit $K, N \to \infty$, $\rho_1 = e^{-\lambda}$ and $\rho_2 = \lambda e^{-\lambda}$ where $\lambda = \ell/c_\epsilon$ for $M = c_\epsilon K$ for any constant c_ϵ.

Lemma 70. For the pruned graph ensemble $\tilde{\mathcal{G}}_\ell(N, K, M)$ the oracle-based peeling decoder fails to peel off atleast $(1-\epsilon)$ fraction of the variable nodes with exponentially decaying probability if $M \geq c_\epsilon K$ where the required c_ϵ and ℓ for various values of ϵ are given in Table. VI.1.

Proof. Instead of reworking the whole proof here from [5], we will list the main steps involved in the proof which we will use further along. Let p_j be the probability that a random defective item is not identified at iteration j of the decoder, in the limit N and $K \to \infty$. Then one can write the density evolution (DE) equations relating p_{j+1} to p_j as

$$p_{j+1} = \left[1 - (\rho_1 + \rho_2(1 - p_j))\right]^{\ell-1}.$$

For this DE, we can see that 0 is not a fixed point and hence $p_j \nrightarrow 0$ as $j \to \infty$. Therefore numerically optimizing the values of c_ϵ and ℓ such that $\lim_{j\to\infty} p_j \leq \epsilon$ gives the optimal values for c_ϵ and ℓ given in Table. VI.1. It was also shown [5, 22] that for such sparse graph systems the actual fraction of the undecoded variable nodes deviates from the average undecoded fraction of the variable nodes given by the DE with exponentially low probability. □

Combining the lemmas and remarks above, the main result from [5] can be summarized as below.

Theorem 71. A random testing matrix from the SAFFRON scheme with $m = 6c_\epsilon K \log_2 N$ tests recovers atleast $(1-\epsilon)$ fraction of the defective items w.h.p of atleast

ϵ	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}
$c_1(\epsilon)$	6.13	7.88	9.63	11.36	13.10	14.84	16.57
ℓ	7	9	10	12	14	15	17

Table VI.1: Constants for various error floor values

$1 - O(\frac{K}{N^2})$. The computational complexity of the decoding scheme is $O(K \log N)$. The constant c_ϵ is given in Table. VI.1 for some values of ϵ.

VI.D Proposed scheme

The main difference between the SAFFRON scheme described in Sec. VI.C and our proposed scheme is that we use a left-and-right-regular sparse-graph instead of left-regular sparse-graph in the first stage for the binning operation.

Definition 72 (Left-and-right-regular sparse graph ensemble)**.** We define $\mathcal{G}_{\ell,r}(N, M)$ to be the ensemble of left-and-right-regular graphs where the $N\ell$ edge connections from the left and $Mr(= N\ell)$ edge connections from the right are paired up according to a permutation π chosen at random from $S_{N\ell}$.

Let $\mathbf{T}_G \in \{0,1\}^{M \times N}$ be the adjacency matrix corresponding to a graph $G \in \mathcal{G}_{\ell,r}(N, M)$ i.e., each column in $\mathbf{T}_G$ corresponding to a variable node has exactly ℓ ones and each row corresponding to a bin node has exactly r ones. And let the universal signature matrix be $\mathbf{U} \in \{0,1\}^{h \times r}$ chosen from the $\mathbf{U}_{r,p}$ ensemble. Then the overall testing matrix $\mathbf{A} := [\mathbf{A}_1^T, \ldots, \mathbf{A}_M^T]^T$ where $\mathbf{A}_i \in \{0,1\}^{h \times N}$ defining the h tests at i^{th} bin is given by

$$\begin{aligned} \mathbf{A}_i &= [\mathbf{0}, \ldots, \mathbf{0}, \mathbf{u}_1, \mathbf{0}, \ldots, \mathbf{u}_2, \mathbf{0}, \ldots, \mathbf{u}_r], \quad \text{where} \\ \mathbf{t}_i &= [0, \ldots, 0,\ 1,\ 0, \ldots,\ 1,\ 0, \ldots,\ 1]. \end{aligned} \tag{VI.2}$$

Note that $\mathbf{A}_i$ is defined by placing the r columns of $\mathbf{U}$ at the r non-zero indices of $\mathbf{t}_i$ and the remaining are padded with zero columns. We can observe that the total number of tests for this scheme is $m = M \times h$ where $h = 2p \log_2 r$.

Example 73. Let us look at an example for $(N, M) = (6, 3)$ and $(\ell, r) = (2, 4)$. Then the adjacency matrix $\mathbf{T}_G$ of a graph $G \in \mathcal{G}_{2,4}(6, 3)$ and a signature matrix $\mathbf{U} \in \{0, 1\}^{4\times 3}$ for $p = 1$ and $\log_2 r = 2$ are given by

$$\mathbf{T}_G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \mathbf{U} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}.$$

Then, the measurement matrix $\mathbf{A}$ with $m = 2pM\lceil \log_2 r \rceil = 12$ tests is given by

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

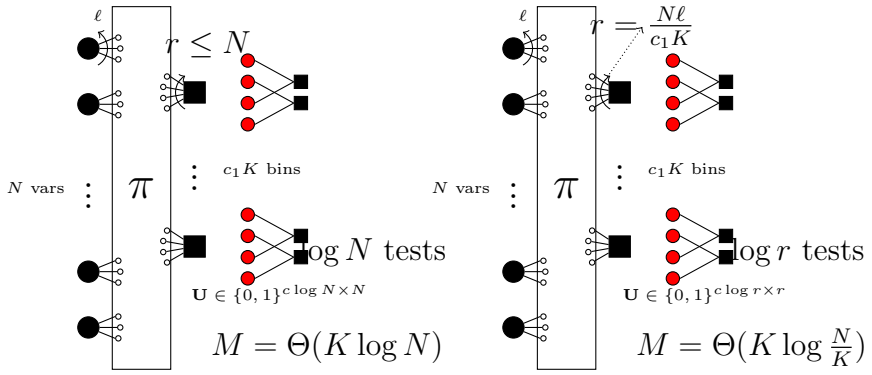

Figure VI.1: Illustration of the main differences between SAFFRON [5] on the left and our regular-SAFFRON scheme on the right. In both the schemes the peeling decoder on sparse graph requires $\Theta(K)$ bins. But for the bin decoder part, in SAFFRON scheme the right degree is a random variable with a maximum value of N and thus requires $\Theta(\log N)$ tests at each bin. Whereas our scheme based on right-regular sparse graph has a constant right degree of $\Theta(\frac{N}{K})$ and thus requires only $\Theta(\log \frac{N}{K})$ tests at each bin. Thus we can improve the number of tests from $\Theta(K \log N)$ to $\Theta(K \log \frac{N}{K})$.

Definition 74 (Regular-SAFFRON)**.** Let the ensemble of testing matrices be $\mathcal{G}_{\ell,r}(N, M) \times \mathbf{U}_{r,p}$ where a graph G from $\mathcal{G}_{\ell,r}(N, M)$ and a signature matrix $\mathbf{U}$ from $\mathbf{U}_{r,p}$ are chosen at random and the testing matrix $\mathbf{A}$ is defined according to Eq. (VI.2). Note that the total number of tests is $2pM \log_2 r$ where $r = \frac{N\ell}{M}$.

For the regular-SAFFRON testing ensemble defined in Def. 74, we employ the iterative decoder described in Sec. VI.C. Similar to the SAFFRON scheme we will analyze the peeling decoder and the bin decoder separately and union bound the total error probability of the decoding scheme. As we have already mentioned the analysis of just the peeling decoder part can be carried out by considering a *simplified oracle-based peeling decoder* on a pruned graph with only the non-zero variable nodes remaining.

Definition 75 (Pruned graph ensemble)**.** We will define the pruned graph ensemble

$\tilde{\mathcal{G}}_{\ell,r}(N,K,M)$ as the set of all graphs obtained from removing a random $N-K$ subset of variable nodes from a graph from left-and-right-regular sparse-graph ensemble $\mathcal{G}_{\ell,r}(N,M)$.

Note that graphs from the pruned ensemble have K variable nodes with a degree ℓ whereas the right degree is not regular anymore.

Lemma 76 (Edge d.d of pruned graph)**.** For the pruned graph ensemble $\tilde{\mathcal{G}}_{\ell,r}(N,K,M)$ it can be shown in the limit $K,N\to\infty$ and $K=o(N)$ that the edge d.d coefficients approach $\rho_1=e^{-\lambda}$ and $\rho_2=\lambda e^{-\lambda}$ where $\lambda=\ell/c$ for the choice of $M=cK$, c being some constant.

Proof. We will first derive $R(x)$ for the pruned graph ensemble and then use the relation $\rho(x)=\frac{R'(x)}{R'(1)}$[22] to derive the edge d.d. Note that all the bin nodes have a uniform degree r before pruning. In the pruning operation we are removing a $N-K$ subset of variable nodes at random which means from the bin node perspective, in an asymptotic sense, this is equivalent to removing each connected edge with a probability $1-\beta$ where $\beta:=\frac{K}{N}$. Under this process the right-node d.d can be written as

$$\begin{aligned} R_1 &= r\beta(1-\beta)^{r-1}, \quad \text{and similarly} \\ R_i &= \binom{r}{i}\beta^i(1-\beta)^{r-i} \;\; \forall i <= r \end{aligned} \tag{VI.3}$$

thus giving us $R(x)=(\beta x+(1-\beta))^r$. This gives us

$$\begin{aligned} \rho(x) &= \frac{r\beta(\beta x+(1-\beta))^{r-1}}{r\beta} \\ &= (\beta x+(1-\beta))^{r-1}. \end{aligned}$$

Thus we can compute that $\rho_1 = (1-\beta)^{r-1}$ and $\rho_2 = (r-1)\beta(1-\beta)^{r-2}$. For $M = cK$ we evaluate these quantities in the limit $K, N \to \infty$ as

$$\begin{aligned}\lim_{K,N\to\infty} \rho_1 &= \lim_{K,N\to\infty} \left(1 - \frac{K}{N}\right)^{\frac{N\ell}{cK}-1} \\ &= e^{-\lambda} \qquad \text{where } \lambda = \frac{\ell}{c}\end{aligned}$$

Similarly we can show $\lim_{K,N\to\infty} \rho_2 = \lambda e^{-\lambda}$. □

Note that even if our initial ensemble is left-and-right-regular the pruned graph ensemble has asymptotically the same degree distribution as in the SAFFRON scheme where the initial ensemble is left-regular.

Lemma 77. For the pruned graph ensemble $\tilde{\mathcal{G}}_{\ell,r}(N, K, M)$ the oracle-based peeling decoder fails to peel off atleast $(1-\epsilon)$ fraction of the variable nodes with exponentially decaying probability for $M = c_\epsilon K$ where ℓ, c_ϵ for various ϵ is given in Table. VI.1.

Proof. We showed in Lemma. 76 that, in the limit of $K, N \to \infty$, the edge degree distribution coefficients ρ_1 and ρ_2 approach the same values as in the SAFFRON scheme (see Lem. 69). Now we follow the exact same approach as that of Lem. 70 where the limiting values of $\rho_1 = e^{-\lambda}$ and $\rho_2 = \lambda e^{-\lambda}$ are used in the DE equations to show that for the given values of ℓ and c_ϵ $\lim_{j\to\infty} p_j \leq \epsilon$. □

Theorem 78. Let $p \in \mathbb{Z}$ such that $K = o(N^{1-1/p})$. A random testing matrix from the proposed regular SAFFRON ensemble $\mathcal{G}_{\ell,\frac{N\ell}{c_\epsilon K}}(N, c_\epsilon K) \times \mathbf{U}_{\frac{N\ell}{c_\epsilon K},p}$ with $m = c \cdot K \log_2 \frac{c_2 N}{K}$ tests recovers atleast $(1-\epsilon)$ fraction of the defective items w.h.p. The computational complexity of the decoding scheme is $\mathcal{O}(K \log \frac{N}{K})$. The constants are $c = 2pc_\epsilon, c_2 = \frac{\ell}{c_\epsilon}$ where ℓ and c_ϵ for various values of ϵ are given in Table. VI.1.

Proof. It remains to be shown that for the proposed regular SAFFRON scheme the total probability of error vanishes asymptotically in K and N. Let E_1 be the event of

oracle-based peeling decoder terminating without recovering atleast $(1-\epsilon)K$ variable nodes. Let E_2 be the event of the bin decoder making an error during the entirety of the peeling process and E_{bin} be the event of one instance of bin decoder making an error. The total probability of error P_e can be upper bounded by

$$\begin{aligned} P_e &\leq \Pr(E_1) + \Pr(E_2) \\ &\leq \Pr(E_1) + K\ell \Pr(E_{\text{bin}}) \\ &\in O\left(\frac{K^p}{N^{p-1}}\right) \end{aligned}$$

where the second inequality is due to the union bound over a maximum of $K\ell$ (number of edges in the pruned graph) instances of bin decoding. The third line is due to the fact that $\Pr(E_1)$ is exponentially decaying in K (see Lemma. 77) and $\Pr(E_{\text{bin}}) = (\frac{c_\epsilon K}{N\ell})^{p-1}$ (see Lemma. 67 and Def. 74) □

VI.E Total recovery: Singleton-only variant

In this section we will look at the proposed regular-SAFFRON scheme but with a decoder that uses only the singleton bins. To elaborate, the only difference is in the decoder which is not iterative in this framework and recovers the variable nodes connected to only the singleton bin nodes and terminates. We will refer to this scheme as *singleton-only* regular-SAFFRON scheme. The trade-off is that we can now recover the *whole* defective set instead of just a large fraction of the defective items with an additional $\log K$ factor tests. Since we do not need to be able to recover resolvable double-tons we only need $2\log_2 r$ number of tests at each bin i.e. we choose $p = 1$ for the signature matrix in Eqn. (VI.1).

Theorem 79. Let $K = o(N)$. For $M = c_\alpha K \log K$ and $(\ell, r) = (c_\alpha \log K, \frac{N}{K})$ a random testing matrix from the regular SAFFRON ensemble $\mathcal{G}_{\ell,r}(N, M) \times \mathbf{U}_{r,1}$ with

$m = 2c_\alpha K \log K \log_2 \frac{N}{K}$ tests the singleton-only decoder fails to recover all the non-zero variable nodes with a vanishing probability of $\mathcal{O}(K^{-\alpha})$ where $c_\alpha = e(1+\alpha)$.

Proof. First we observe that for the choice of $(\ell, r) = (c_\alpha \log K, \frac{N}{K})$ number of bins $M = \frac{N\ell}{r} = c_\alpha K \log K$ and the number of tests in each bin is $2\log_2 \frac{N}{K}$. From Lem. 67 we know that a singleton bin is guaranteed to be decoded by the bin decoder. Thus it is enough if we show that for this choice for the number of bins M all the variable nodes in the pruned graph are connected to atleast one singleton bin w.h.p of $1 - \mathcal{O}(K^{-\alpha})$.

In the pruned graph ensemble, for any particular variable node, the probability that any of the ℓ connected bit nodes are not a singleton can be given by $(1-R_1)^\ell$ where R_1 is the probability that a bin node in the pruned graph ensemble is a singleton. In the limit $K, N \to \infty$ the value of R_1 approaches (from Eq. VI.3)

$$\begin{aligned} R_1 &= \lim_{K,N\to\infty} r\beta(1-\beta)^{r-1} \\ &= \lim_{\frac{N}{K}\to\infty} \left(1 - \frac{K}{N}\right)^{\frac{N}{K}-1} \\ &= e^{-1} \end{aligned}$$

By using union bound over all the K variable nodes in the pruned graph, the probability P_e that the singleton-only decoder fails to recover a defective item can be

bounded by

$$\begin{aligned} P_e &\leq K(1-R_1)^{\ell} \\ &= \mathcal{O}\left(K(1-e^{-1})^{e(1+\alpha)\log K}\right) \\ &= \mathcal{O}\left(Ke^{-e^{-1}e(1+\alpha)\log K}\right) \\ &= \mathcal{O}\left(K^{-\alpha}\right). \end{aligned}$$

In third line we used $(1-x) \leq e^{-x}$. □

VI.F Robust group testing

In this section we extend our scheme to the group testing problem where the test results can be noisy. Formally, the signal model can be described as

$$\mathbf{y} = \mathbf{A} \odot \mathbf{x} + \mathbf{w},$$

where $\mathbf{w} \in \{0,1\}^N$ is an i.i.d. noise vector distributed according to Bernoulli distribution with parameter $0 < q < \frac{1}{2}$ and the addition is over binary field.

Testing scheme

In [5] for the robust group testing problem, the signature matrix used for noiseless group testing problem is modified using an error control code such that it can handle singletons and resolvable doubletons in the presence of noise. The binning operation as defined by the bipartite graph is exactly identical to that of noiseless case. We describe the modifications to the signature matrix and the bin detection decoding scheme as given in [5] for the sake of completeness and then state the performance bounds for our scheme for the noisy group testing problem.

Let $\mathcal{C}_n$ be a binary error-correcting code with the following definition:

- Let the encoder and decoder functions be $f : \{0,1\}^n \rightarrow \{0,1\}^{\frac{n}{R}}$ and $g : \{0,1\}^{\frac{n}{R}} \rightarrow \{0,1\}^n$ respectively where R is the rate of the code.

For ease of analysis and tight upper bound for the number of tests we will use random codes and the optimal maximum-likelihood decoder which gives us the properties:

- There exists a sequence of codes $\{\mathcal{C}_n\}$ with the rate of each code being R satisfying

$$R < 1 - H(q) - \delta = 1 + q\log_2 q + \overline{q}\log_2 \overline{q} - \delta \tag{VI.4}$$

  for any arbitrary small constant δ such that the probability of error $\Pr\left(g(\mathbf{x}+\mathbf{w}) \neq \mathbf{x}\right) < 2^{-\kappa n}$ for some $\kappa > 0$. In Eqn. VI.4, $\overline{q} := 1 - q$.

Even though the computational complexity of using random codes is exponential in block length of the code since the block length for our application is $\mathcal{O}(\log \frac{N}{K})$ and hence we have an overall computational complexity of $\mathcal{O}(N)$. But in practice one can use any of the popular error-correcting codes such as spatially-coupled LDPC codes or polar codes which are known to be capacity achieving [31, 58] whose computational complexity is linear in block length.

The modified signature matrix $\mathbf{U}'_{r,p}$ can be described via $\mathbf{U}_{r,p}$ given in Eq. (VI.1)

and encoding function f for $\mathcal{C}_n$ where $n = \lceil \log_2 r \rceil$ as follows:

$$\mathbf{U}'_{r,p} := \begin{bmatrix} f(\mathbf{b}_1) & f(\mathbf{b}_2) & \cdots & f(\mathbf{b}_r) \\ \overline{f(\mathbf{b}_1)} & \overline{f(\mathbf{b}_2)} & \cdots & \overline{f(\mathbf{b}_r)} \\ f(\mathbf{b}_{\pi_1^1}) & f(\mathbf{b}_{\pi_2^1}) & \cdots & f(\mathbf{b}_{\pi_r^1}) \\ \overline{f(\mathbf{b}_{\pi_1^1})} & \overline{f(\mathbf{b}_{\pi_2^1})} & \cdots & \overline{f(\mathbf{b}_{\pi_r^1})} \\ \cdots & & \vdots & \\ f(\mathbf{b}_{\pi_1^{p-1}}) & f(\mathbf{b}_{\pi_2^{p-1}}) & \cdots & f(\mathbf{b}_{\pi_r^{p-1}}) \\ \overline{f(\mathbf{b}_{\pi_1^{p-1}})} & \overline{f(\mathbf{b}_{\pi_2^{p-1}})} & \cdots & \overline{f(\mathbf{b}_{\pi_r^{p-1}})} \end{bmatrix} \tag{VI.5}$$

Then the overall testing matrix $\mathbf{A}$ is defined in identical fashion to the definition in Sec. VI.C for the case of noiseless case except that $\mathbf{U}$ will be replaced by $\mathbf{U}'$ in Eqn. (VI.5). Formally it can be defined as $\mathbf{A} := [\mathbf{A}_1^T, \ldots, \mathbf{A}_{M_1}^T]^T$ where $\mathbf{A}_i = \mathbf{U}' \operatorname{diag}(\mathbf{t}_i)$ where the binary vectors $\mathbf{t}_i$ are defined in Sec. VI.C.

Decoding

The decoding scheme for the robust group testing, similar to the case of noiseless case, has two parts with the peeling part of the decoder identical to that of the noiseless case whereas the bin detection part differs slightly with an extra step of decoding for the error control code involved.

Given the test output vector at a bin $\mathbf{y} = [\mathbf{y}_{01}^T, \mathbf{y}_{02}^T, \mathbf{y}_{11}^T, \ldots, \mathbf{y}_{(p-1)2}^T]^T$, the bin detection for the noisy case can be summarized as following: The decoder $\forall i \in [0 : p-1]$ applies the decoding function $g(\cdot)$ to the first segment $\mathbf{y}_{i1}$ in each section i and obtains the location l_i whose binary expansion is equal to the error-correcting decoder output $g(\mathbf{y}_{i1})$. The decoder then declares the bin as a singleton if $\pi_{l_0}^i = l_i \; \forall i$.

Similarly given that one of the variable nodes connected to the bin is already decoded to be non-zero, the resolvable double-ton decoding can be summarized as

following. Let the location of the already recovered variable node in the bin (originally a double-ton) be l_0 then the test output can be given as

$$\begin{bmatrix} \mathbf{y}_{01} \\ \mathbf{y}_{02} \\ \mathbf{y}_{11} \\ \vdots \\ \mathbf{y}_{p2} \end{bmatrix} = \mathbf{u}_{l_0} \vee \mathbf{u}_{l_1} + \mathbf{w} = \begin{bmatrix} f(\mathbf{b}_{l_0}) \\ \overline{f(\mathbf{b}_{l_0})} \\ f(\mathbf{b}_{\pi^1_{l_0}}) \\ \vdots \\ \overline{f(\mathbf{b}_{\pi^p_{l_0}})} \end{bmatrix} \vee \begin{bmatrix} f(\mathbf{b}_{l_1}) \\ \overline{f(\mathbf{b}_{l_1})} \\ f(\mathbf{b}_{\pi^1_{l_1}}) \\ \vdots \\ \overline{f(\mathbf{b}_{\pi^p_{l_1}})} \end{bmatrix} + \begin{bmatrix} \mathbf{w}_{01} \\ \mathbf{w}_{02} \\ \mathbf{w}_{11} \\ \vdots \\ \mathbf{w}_{p2} \end{bmatrix}$$

where the location of the second non-zero variable node l_1 needs to be recovered. Given $\mathbf{y} = \mathbf{u}_{l_0} \vee \mathbf{u}_{l_1} + \mathbf{w}$ and $\mathbf{u}_{l_0}$, the first segments of each section in $\mathbf{u}_{l_1} + \mathbf{w}$ can be recovered since for each segment of $\mathbf{u}_{l_0}$ either the vector $f(\mathbf{b}_{\pi^k_{l_0}})$ or it's complement is available. Once the first section $f(\mathbf{b}_{\pi^i_{l_1}}) + \mathbf{w}$ of each segment i is recovered, we apply singleton decoding procedure and rules as described above.

Lemma 80 (Robust Bin Decoder Analysis)**.** For a signature matrix $\mathbf{U}'_{r,p}$ as described in (VI.5), the robust bin decoder misses a singleton with probability no greater than $\frac{p}{r^\kappa}$. The robust bin decoder wrongly declares a singleton with probability no greater than $\frac{1}{r^{p\kappa+p-1}}$.

Proof. Let E_i be the event that the error-control decoder $g(\mathbf{y}_{i1})$ commits an error at section i. From Eqn. (VI.4) we know that $\Pr(E_i) = 2^{-\kappa \log r} = r^{-\kappa}$. The robust bin decoder misses a singleton if the error-control decoder $g(\mathbf{y}_{i1})$ commits an error at any one section. Thus the probability of missing a singleton can be upper bounded by applying union bound over all the sections $i \in [0 : p-1]$ giving the required result.

Consider a singleton bin and let the event where the robust bin decoder outputs a singleton hypothesis but the wrong index be E_{bin}. This event happens when the error-control decoder commits an error and outputs the exact same wrong index

at each and every section. We assume that given the error-control decoder makes an error, the output is uniformly random among all the remaining indices. Thus $\Pr(E_{\text{bin}})$ can be upper bounded by $\frac{1}{r^\kappa}(\frac{1}{r^{1+\kappa}})^{p-1}$ which upon simplification gives us the required result. □

The fraction of missed singletons can be compensated by using $M(1+\frac{p}{r^\kappa})$ instead of M such that the total number of singletons decoded will be $M(1+\frac{p}{r^\kappa})(1-\frac{p}{r^\kappa}) \approx M$.

Theorem 81. Let $p \in \mathbb{Z}$ such that $K = o\left(N^{1-1/p}\right)$. The proposed robust regular SAFFRON scheme using $m = c \cdot K \log_2 \frac{N\ell}{c_\epsilon K}$ tests recovers atleast $(1-\epsilon)$ fraction of the defective items w.h.p. where $c = 2p\beta(q)c_\epsilon$ and $\beta(q) = 1/R$.

Proof. Similar to the noiseless case the total probability of error P_e is dominated by the performance of bin decoder.

$$\begin{aligned} P_e &\leq \Pr(E_1) + K\ell \Pr(E_{\text{bin}}) \\ &= \Pr(E_1) + \mathcal{O}\left(\frac{K^{p+p\kappa}}{N^{p-1+p\kappa)}}\right) \\ &= \mathcal{O}\left(\frac{N^{(p-1)(1+\kappa)}}{N^{p-1+p\kappa)}}\right) \\ &\in \mathcal{O}(N^{-\kappa}) \end{aligned}$$

where the second line is due to Lem. 80 and the third line is due to the fact that $\Pr(E_1)$ is exponentially decaying in K and $K \leq N^{(p-1)/p}$ for large enough K, N. □

VI.G Simulation results

In this section we will evaluate the performance of the proposed regular-SAFFRON scheme via Monte Carlo simulations and compare it with the results of SAFFRON scheme provided in [5] for both the noiseless and noisy models.

Noiseless group testing

As per Thm. 78 the proposed regular SAFFRON scheme requires only $6c_\epsilon K \log \frac{N\ell}{c_\epsilon K}$ tests as opposed to $6c_\epsilon K \log N$ tests of SAFFRON scheme to recover $(1-\epsilon)$ fraction of defective items with a high probability. We demonstrate this by simulating the performance for the system parameters summarized below.

- We fix $N = 2^{16}$ and $K = 100$
- For $\ell \in \{3, 5, 7\}$ we vary the number of bins $M = cK$.
- In Eqn. VI.1 the parameter $p = 2$ is chosen for matrix $\mathbf{U}$
- Thus the bin detection size is $h = 6 \log_2 \frac{N\ell}{cK}$
- Hence the total number of tests $m = 6cK \log_2 \left(\frac{N\ell}{cK}\right)$

The results are shown in Fig. VI.2. We observe that there is clear improvement in performance for the proposed regular SAFFRON scheme when compared to the SAFFRON scheme for each $\ell \in \{3, 5, 7\}$.

Noisy group testing

Similar to the noiseless group testing problem we simulate the performance of our robust regular-SAFFRON scheme and compare it with that of the SAFFRON scheme. For convenience of comparison we choose our system parameters identical to the choices in [5]. The system parameters are summarized below:

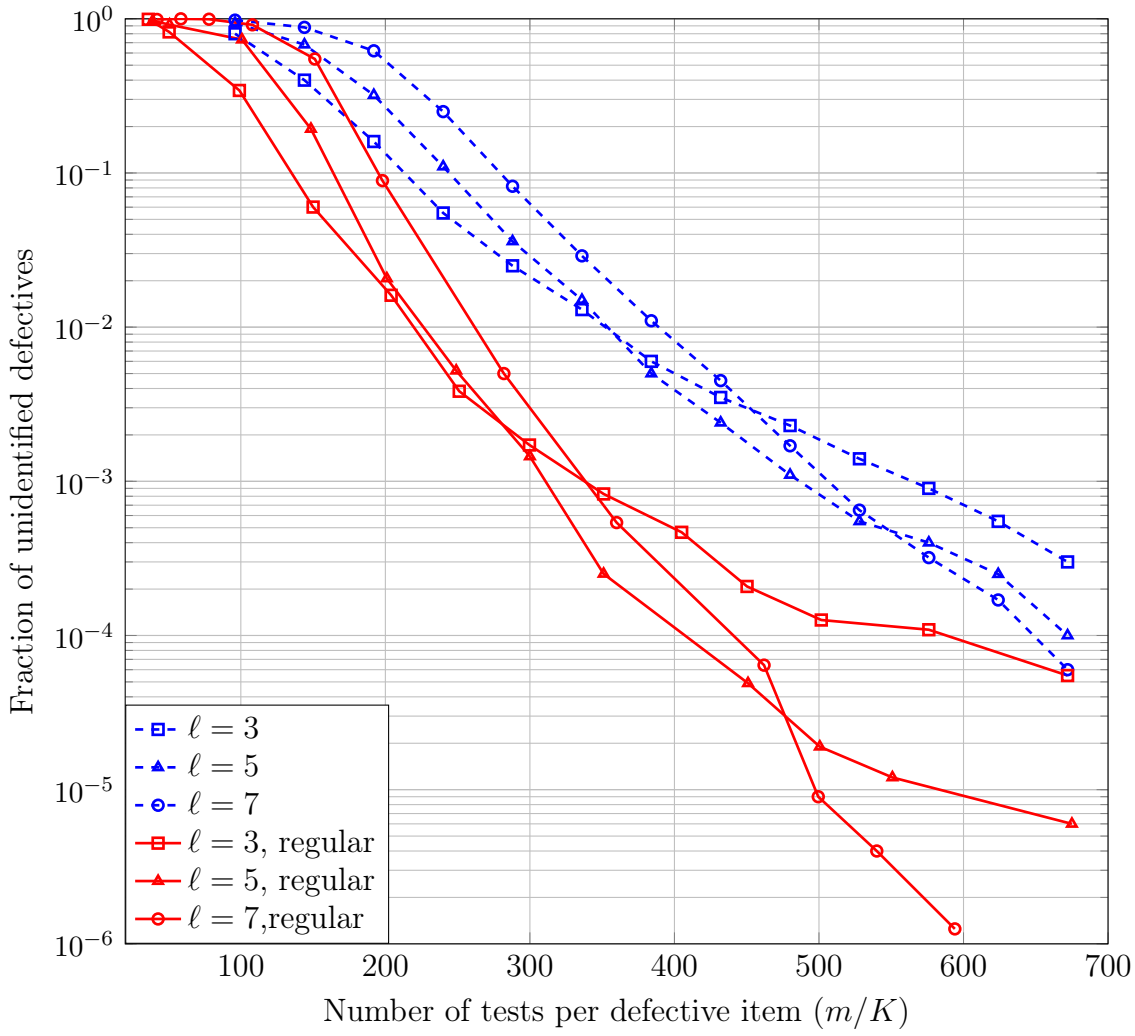

Figure VI.2: MonteCarlo simulations for $K = 100, N = 2^{16}$. We compare the SAFFRON scheme [5] with the proposed regular SAFFRON scheme for various left degrees $\ell \in \{3, 5, 7\}$. The plots in blue indicate the SAFFRON scheme and the plots in red indicate our regular SAFFRON scheme based on left-and-right-regular bipartite graphs.

- $N = 2^{32}, K = 2^7$. We fix $\ell = 12, M = 11.36K$
- BSC noise parameter $q \in \{0.03, 0.04, 0.05\}$
- In Eqn. VI.1 the parameter $p = 1$ is chosen for matrix $\mathbf{U}$

- Thus the bin detection size is $h = 4\log_2 \frac{N\ell}{M}$

The results are shown in Fig. VI.3. Note that for the above set of parameters the right degree $r = \frac{N\ell}{M} \approx 26$. We choose to operate in field $GF(2^7)$ thus giving us a message length of 4 symbols. For the choice of code we use a $(4+2e, 4)$ Reed-Solomon code for $e \in [0:8]$ thus giving us a column length of $4 \times 7(4+2e)$ bits at each bin and the total number of tests $m = 28M(4+2e)$.

VI.H Conclusion

We addressed the Group Testing problem of identifying K defective items out of N items and proposed a new construction for the testing matrix based on left-*and*-right-regular sparse-graph codes. It was shown that this improves the testing complexity upon the previous results for the approximate version of the Group Testing problem and achieves asymptotically vanishing error probability under sub-linear time, order optimal, computational complexity. It was also shown that the proposed scheme with a variant of the original decoder has a testing complexity that is only $\log K$ factor away from the lower bound for the probabilistic version of the Group Testing problem with order optimal computational complexity. In the non-asymptotic regime, it was demonstrated through numerical simulations that the proposed scheme improves upon the existing sparse-graph based schemes in terms of the number of tests required to achieve a fixed target error probability.

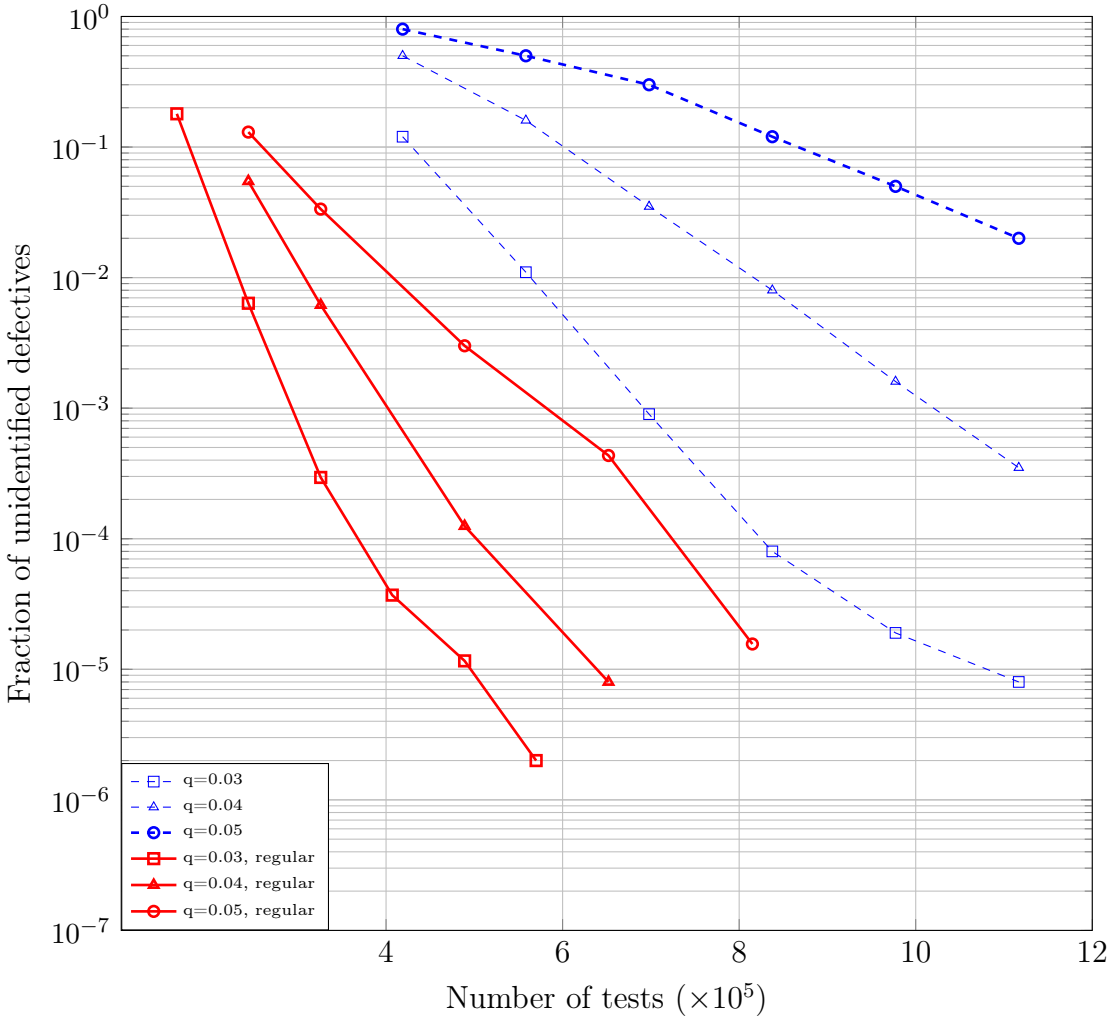

Figure VI.3: MonteCarlo simulations for $K = 128, N = 2^{32}$. We compare the SAFFRON scheme with the proposed regular-SAFFRON scheme for a left degree $\ell = 12$. We fix the number of bins and vary the rate of the error control code used. The plots in blue indicate the SAFFRON scheme[5] and the plots in red indicate the regular-SAFFRON scheme based on left-and-right-regular bipartite graphs.

VII. CONCLUSIONS AND FUTURE DIRECTIONS

In this thesis we provided solutions to some problems in massive multiple access and sparse signal recovery using tools from coding theory. However we believe that there are a wide variety of applications with huge potential in applying these coding theory tools. Below, we list some of the questions that emanate from this thesis that need to be pursued and also a few potential applications of the solution designs discussed.

- Consider the compressed sensing problem studied in Ch. II:

$$\vec{\mathrm{y}} = \mathbf{A}\vec{\mathrm{b}} + \vec{\mathrm{z}},$$

  where the non-zero elements of the T-sparse vector $\vec{\mathrm{b}}$ are all equal to one and the sparsity is very small, $T \in [1 : 10]$. This specific compressed sensing problem is not extensively studied in the literature for the non-asymptotic regime. Although we derived some new bounds on T-disjunctive codes as an application for the sensing matrix, a full characterization of the sensing matrix suitable for this problem warrants further study.

- In Ch. III, given a probability distribution for the repetition pattern of each user in the random multiple access problem, analytic expressions to compute the error probability of peeling decoder are derived. Based on these analytic expressions, an iterative linear programming optimization technique based on first order approximations to error probability, similar to [3], need to be studied. Through this approach distributions can be found which, for number of users $n = 1000$, can potentially achieve values of throughput larger than the current

best known value $\approx 80\%$.

- In Ch. IV lattice construction based on nested linear spatially coupled LDPC code ensembles is proposed. It was shown that the proposed lattices are optimal for the unconstrained AWGN channel i.e., *Poltyrev-good*. Given such *Poltyrev-good* lattices, it was shown recently [59, 60] that applying appropriate discrete Gaussian shaping over the lattice so that the power constraint is satisfied, the capacity of the power constrained AWGN channel can be achieved. The optimality of the low complexity multi-level decoding considered for the proposed SC-LDPC lattices in Ch. IV, under the discrete Gaussian shaping needs to be studied. If this issue can be resolved affirmatively, the capacity of the three user symmetric interference channel can be achieved by the proposed lattices, overcoming the demonstrated 1.53dB gap in Sec. IV.C.4, due to hyper cube shaping.

- In Chapters V & VI we modified the earlier sensing schemes due to Ramchandran *et al.,* by replacing the left-regular with left-and-right-regular bipartite graphs. This not only enabled us to derive sharper results in the asymptotic regime matching the lower bounds but also demonstrated improved performance in the non-asymptotic regime. A thorough comparison, in terms of the measurement and computational complexities, with the popular schemes in the literature for support recovery and group testing, particularly in the non-asymptotic regime needs to be undertaken.

REFERENCES

[1] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 803–834, 2011.

[2] Y. Polyanskiy, "A perspective on massive random-access," in *Proceedings of the IEEE International Symposium on Information Theory*, pp. 2523–2527, 2017.

[3] A. Amraoui, A. Montanari, and R. Urbanke, "How to find good finite-length codes: from art towards science," *Transactions on Emerging Telecommunications Technologies*, vol. 18, no. 5, pp. 491–508, 2007.

[4] X. Li, S. Pawar, and K. Ramchandran, "Sub-linear time compressed sensing using sparse-graph codes." `http://www.eecs.berkeley.edu/~xiaoli/TR_CS_sublinear.pdf`.

[5] K. Lee, R. Pedarsani, and K. Ramchandran, "SAFFRON: A fast, efficient, and robust framework for group testing based on sparse-graph codes," *arXiv preprint arXiv:1508.04485*, 2015.

[6] N. Abramson, "The ALOHA system: another alternative for computer communications," in *Proceedings of the November 17-19, 1970, fall joint computer conference*, pp. 281–285, ACM, 1970.

[7] E. Casini, R. De Gaudenzi, and O. D. R. Herrero, "Contention resolution diversity slotted ALOHA (CRDSA): An enhanced random access schemefor satellite access packet networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 4, 2007.

[8] G. Liva, "Graph-based analysis and optimization of contention resolution diversity slotted ALOHA," *IEEE Transactions on Communications*, vol. 59, no. 2, pp. 477–487, 2011.

[9] K. R. Narayanan and H. D. Pfister, "Iterative collision resolution for slotted ALOHA: An optimal uncoordinated transmission policy," in *Proceedings of 7th International Symposium on Turbo Codes and Iterative Information Processing*, pp. 136–139, 2012.

[10] O. Ordentlich and Y. Polyanskiy, "Low complexity schemes for the random access gaussian channel," in *Proceedings of the IEEE International Symposium on Information Theory*, pp. 2528–2532, 2017.

[11] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[12] A. Carleial, "Interference channels," *IEEE Transactions on Information Theory*, vol. 24, no. 1, pp. 60–70, 1978.

[13] H. Sato, "The capacity of the gaussian interference channel under strong interference (corresp.)," *IEEE transactions on information theory*, vol. 27, no. 6, pp. 786–788, 1981.

[14] R. H. Etkin, N. David, and H. Wang, "Gaussian interference channel capacity to within one bit," *IEEE Transactions on Information Theory*, vol. 54, no. 12, pp. 5534–5562, 2008.

[15] S. Sridharan, A. Jafarian, S. Vishwanath, and S. A. Jafar, "Capacity of symmetric k-user Gaussian very strong interference channels," in *Global Telecommunications Conference, 2008.*, pp. 1–5, IEEE, 2008.

[16] X. Li, S. Pawar, and K. Ramchandran, "Sub-linear time compressed sensing using sparse-graph codes," in *Proceedings of the IEEE International Symposium on Information Theory*, pp. 1645–1649, 2015.

[17] R. Dorfman, "The detection of defective members of large populations," *The Annals of Mathematical Statistics*, vol. 14, no. 4, pp. 436–440, 1943.

[18] B. Rimoldi and R. Urbanke, "A rate-splitting approach to the Gaussian multiple-access channel," vol. 42, no. 2, pp. 364–375, 1996.

[19] L. Ping, L. Liu, K. Wu, and W. K. Leung, "Interleave division multiple-access," *IEEE Transactions on Wireless Communications*, vol. 5, no. 4, pp. 938–947, 2006.

[20] J. Zhan, B. Nazer, U. Erez, and M. Gastpar, "Integer-forcing linear receivers," *IEEE Transactions on Information Theory*, vol. 60, no. 12, pp. 7661–7685, 2014.

[21] P. Mathys, "A class of codes for a t active users out of n multiple-access communication system," *IEEE Transactions on Information Theory*, vol. 36, no. 6, pp. 1206–1219, 1990.

[22] T. Richardson and R. Urbanke, *Modern coding theory*. Cambridge University Press, 2008.

[23] P. Z. Fan, M. Darnell, and B. Honary, "Superimposed codes for the multiaccess binary adder channel," *IEEE Transactions on Information Theory*, vol. 41, no. 4, pp. 1178–1182, 1995.

[24] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, "Practical loss-resilient codes," in *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pp. 150–159, ACM, 1997.

[25] U. Erez, S. Litsyn, and R. Zamir, "Lattices which are good for (almost) everything," *IEEE Transactions on Information Theory*, vol. 51, no. 10, pp. 3401–3416, 2005.

[26] R. Zamir, *Lattice Coding for Signals and Networks: A Structured Coding Approach to Quantization, Modulation, and Multiuser Information Theory*. Cambridge University Press, 2014.

[27] A. Vem, Y.-C. Huang, K. R. Narayanan, and H. D. Pfister, "Multilevel lattices based on spatially-coupled LDPC codes with applications," in *Proceedings of the IEEE International Symposium on Information Theory*, pp. 2336–2340, 2014.

[28] E. Barnes and N. Sloane, "New lattice packings of spheres," *Canad. J. Math*, vol. 35, pp. 117–130, 1983.

[29] G. D. Forney Jr, M. D. Trott, and S.-Y. Chung, "Sphere-bound-achieving coset codes and multilevel coset codes," *IEEE Transactions on Information Theory*, vol. 46, no. 3, pp. 820–850, 2000.

[30] S. Kudekar, T. Richardson, and R. Urbanke, "Spatially coupled ensembles universally achieve capacity under belief propagation," in *Proceedings of the IEEE International Symposium on Information Theory*, pp. 453–457, 2012.

[31] S. Kumar, A. J. Young, N. Macris, and H. D. Pfister, "Threshold saturation for spatially coupled LDPC and LDGM codes on BMS channels," *IEEE Transactions on Information Theory*, vol. 60, no. 12, pp. 7389–7415, 2014.

[32] Y. Yan, C. Ling, and X. Wu, "Polar lattices: where Arıkan meets Forney," in *Proceedings of the IEEE International Symposium on Information Theory*, pp. 1292–1296, 2013.

[33] S. A. Jafar and S. Vishwanath, "Generalized degrees of freedom of the symmetric Gaussian k-user interference channel," *arXiv preprint arXiv:0804.4489*, 2008.

[34] M. C. Estela, C. Ling, and J.-C. Belfiore, "Barnes-wall lattices for the symmetric interference channel," in *Proceedings of the IEEE International Symposium on Information Theory*, pp. 2239–2243, 2013.

[35] G. Poltyrev, "On coding without restrictions for the AWGN channel," *IEEE Transactions on Information Theory*, vol. 40, no. 2, pp. 409–417, 1994.

[36] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured codes," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6463–6486, 2011.

[37] J. Conway and N. Sloane, *Sphere Packings, Lattices and Groups*. Springer-Verlag, New York, 1988.

[38] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC," *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 803–834, 2011.

[39] M.-R. Sadeghi, A. H. Banihashemi, and D. Panario, "Low-density parity-check lattices: construction and decoding analysis," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4481–4495, 2006.

[40] A. H. Banihashemi and F. R. Kschischang, "Tanner graphs for group block codes and lattices: construction and complexity," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 822–834, 2001.

[41] M. J. Wainwright, "Information-theoretic limits on sparsity recovery in the high-dimensional and noisy setting," *Information Theory, IEEE Transactions on*,

vol. 55, no. 12, pp. 5728–5741, 2009.

[42] A. Vem, N. T. Janakiraman, and K. Narayanan, "Sub-linear time compressed sensing for support recovery using left and right regular sparse-graph codes," in *Proceedings of the IEEE Information Theory Workshop*, pp. 429–433, 2016.

[43] S. Jafarpour, W. Xu, B. Hassibi, and R. Calderbank, "Efficient and robust compressed sensing using optimized expander graphs," *Proceedings of the IEEE International Symposium on Information Theory*, vol. 55, no. 9, pp. 4299–4308, 2009.

[44] D.-Z. Du and F. K. Hwang, *Combinatorial group testing and its applications*, vol. 12. World Scientific, 1999.

[45] H.-B. Chen and F. K. Hwang, "A survey on nonadaptive group testing algorithms through the angle of decoding," *Journal of Combinatorial Optimization*, vol. 15, no. 1, pp. 49–59, 2008.

[46] D. M. Malioutov and K. R. Varshney, "Exact rule learning via boolean compressed sensing.," in *Proceedings of International Conference on Machine Learning*, pp. 765–773, 2013.

[47] M. T. Goodrich, M. J. Atallah, and R. Tamassia, "Indexing information for data forensics," in *International Conference on Applied Cryptography and Network Security*, pp. 206–221, Springer, 2005.

[48] A. Emad and O. Milenkovic, "Poisson group testing: A probabilistic model for nonadaptive streaming boolean compressed sensing," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3335–3339, IEEE, 2014.

[49] C. L. Chan, S. Jaggi, V. Saligrama, and S. Agnihotri, "Non-adaptive group testing: Explicit bounds and novel algorithms," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 3019–3035, 2014.

[50] G. K. Atia and V. Saligrama, "Boolean compressed sensing and noisy group testing," *IEEE Transactions on Information Theory*, vol. 58, no. 3, pp. 1880–1901, 2012.

[51] A. G. D'yachkov and V. V. Rykov, "Bounds on the length of disjunctive codes," *Problemy Peredachi Informatsii*, vol. 18, no. 3, pp. 7–13, 1982.

[52] P. Erdös, P. Frankl, and Z. Füredi, "Families of finite sets in which no set is covered by the union ofr others," *Israel Journal of Mathematics*, vol. 51, no. 1, pp. 79–89, 1985.

[53] W. Kautz and R. Singleton, "Nonrandom binary superimposed codes," *IEEE Transactions on Information Theory*, vol. 10, no. 4, pp. 363–377, 1964.

[54] E. Porat and A. Rothschild, "Explicit nonadaptive combinatorial group testing schemes," *IEEE Transactions on Information Theory*, vol. 57, no. 12, pp. 7982–7989, 2011.

[55] P. Indyk, H. Q. Ngo, and A. Rudra, "Efficiently decodable non-adaptive group testing," in *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pp. 1126–1142, Society for Industrial and Applied Mathematics, 2010.

[56] A. Mazumdar, "Nonadaptive group testing with random set of defectives," *arXiv preprint arXiv:1503.03597*, 2015.

[57] A. Vem, N. T. Janakiraman, and K. R. Narayanan, "Group testing using left-and-right-regular sparse-graph codes," *arXiv preprint arXiv:1701.07477*, 2017.

[58] S. Kudekar, T. Richardson, and R. L. Urbanke, "Spatially coupled ensembles universally achieve capacity under belief propagation," *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 7761–7813, 2013.

[59] C. Ling and J.-C. Belfiore, "Achieving awgn channel capacity with lattice gaussian coding," *IEEE Transactions on Information Theory*, vol. 60, no. 10, pp. 5918–5929, 2014.

[60] Y. Yan, L. Liu, C. Ling, and X. Wu, "Construction of capacity-achieving lattice codes: Polar lattices," *arXiv preprint arXiv:1411.0187*, 2014.